GRADO DE INGENIERÍA DE SISTEMAS AUDIOVISUALES

Universidad Carlos III de Madrid

TRABAJO FIN DEL GRADO

shopC: JUEGO EDUCATIVO DE TABLERO SOPORTADO POR ORDENADOR BASADO EN CENTRO COMERCIAL

AUTOR: MARÍA JULIÁN MATEOS

TUTOR: PEDRO J. MUÑOZ MERINO

Leganés, marzo 2013

Agradecimientos:

Estas son mis últimas líneas como estudiante del Grado de Ingeniería de Sistemas Audiovisuales. Atrás dejo muchos años maravillosos y a la vez duros. Dejo profesores, compañeros y exámenes; y me llevo amigos y experiencias que me ayudaran mucho en mi camino.

Solo me quedan buenos recuerdos de la Universidad y eso solo se puede deber a ellos, gente que he conocido en la Carlos III y que se han convertido en mis amigos, con los que he sufrido y reído estos años; estoy segura que sin ellos no estaría escribiendo estas líneas. Solo por conocerlos esto ha merecido la pena: Belén, Jorge, Paloma, María Lozano, Mario, Manu, Javi, Biko y en especial a Natalia, Diego y Susana, que han estado conmigo a diario, en lo bueno y en lo malo. Su amistad es lo mejor que me llevo de estos años.

Por supuesto, que agradecer a mis 'cone-padres', Reyes y Pichaque, mis compañeros de piso. Ellos han sido como mi familia en Leganés y nunca olvidaré todos los momentos que hemos pasado juntos. Gracias por hacerme sentir como en casa.

También me gustaría agradecer a mis amigas del pueblo, Las Castañazos, y a mis Kikis, Leti, Juli, Cristina y Vero, que me han ayudado siempre y aunque en la distancia, también han formando parte de esta etapa de mi vida.

Pero mi agradecimiento más infinito es para mis padres, Elisa y Valentín, que me han dado todo lo que tengo y todo lo que soy y que sin ellos nada hubiera sido posible. Son lo más importante de mi vida, junto a mi hermano, Gonzalo, y en general a toda mi familia, que están siempre apoyándome.

Por último, quiero dar las gracias a mi tutor, Pedro, por su gran ayuda y a todas esas personas que para bien o para mal han estado en mi camino estos seis años, pues de todos he aprendido algo.

Gracias a todos de corazón.

(Hoja en blanco)

Resumen

Los juegos educativos están cada vez más valorados en el ámbito de la educación, y se utilizan habitualmente como un complemento para los alumnos de manera que puedan ampliar y demostrar los conocimientos adquiridos en las clases. Los juegos para la educación tienen la finalidad de motivar y divertir a los estudiantes al mismo tiempo que aprenden.

En este trabajo se ha desarrollado un programa informático que es un juego de tablero. Se utiliza el lenguaje de programación Java y el objetivo es que los jugadores/estudiantes puedan al mismo tiempo aprender y divertirse.

El juego soportado por ordenador consta de un tablero que imita a un centro comercial donde cada casilla es una tienda. El jugador tirará un dado en cada turno y podrá comprar la tienda donde ha caído respondiendo a unas preguntas; dependiendo de los aciertos pagará una cantidad de dinero u otra. Cuando un jugador caiga en una casilla que esté comprada por otro jugador, deberá contestar a otra pregunta y pagará dependiendo si acierta o no. Si es una tienda comprada por él, se le pagará un tanto por ciento del precio de la tienda. El juego termina cuando uno de los jugadores se queda sin dinero o si han pasado 50 turnos. El ganador del juego será el jugador que más dinero tenga en ese momento y el perdedor el que menos.

Además, el juego ha sido evaluado, utilizando métodos cuantitativos, por un grupo de alumnos del Grado de Ingeniería de Sistemas Audiovisuales que han interaccionado con el juego al que se ha incluido preguntas relacionadas con la asignatura Arquitectura de Sistemas. Para la evaluación se ha tomado como base y adaptado un cuestionario utilizado en los años 2011 y 2012 en la asignatura de Laboratorio de Arquitectura de Ordenadores para la evaluación de un software de competición.

Palabras clave

Juegos, educación, aprendizaje soportado por ordenador, teleeducación, preguntas, Java, MySQL, SQL, base de datos, Eclipse, XAMPP, programación Java.

(Hoja en blanco)

ÍNDICE

Índice de figuras

Índice de tablas

(Hoja en blanco)

1. Introducción

Para aprender hay que estudiar y esto es muy duro en ocasiones. Todo sería más fácil si hiciéramos del saber una diversión, si pudiéramos mezclar el aprendizaje y pasarlo bien.

Aunque en muchas ocasiones es muy cansado y poco divertido, una de las maneras tradicionales de adquirir conocimientos es coger un libro y pasar horas y horas delante de él. Sin embargo, se puede motivar a los alumnos a estudiar, y ayudarles a ampliar y aplicar sus conocimientos de una forma atractiva. Se trata de que el estudiante aprenda a partir de sus conocimientos y capacidades, que mejore su socialización y qué, rompiendo con la rutina de las clases docentes aumente su interés por las asignaturas.

Para el docente también es difícil, en muchas ocasiones está falto de medios para lograr que el alumno se motive y desarrolle un pensamiento lógico.

La idea es que implementando herramientas innovadoras, se capte la atención de los estudiantes aumentando los deseos de participar activamente en las actividades de las clases y aumentando también sus ganas de aprender. Se quiere provocar dos efectos: divertirse y aprender a la vez, y un juego educativo puede desempeñar estas funciones.

Los juegos han acompañado al hombre toda su vida, son una herramienta fundamental de aprendizaje que sin darnos cuenta utilizamos desde pequeños y que nos aporta muchos beneficios. Los bebés aprenden jugando sobre el mundo que les rodea y los niños juegan imitando el mundo adulto e interiorizando cómo será su vida en unos años, aprendiendo a resolver problemas para desenvolverse por sí mismos en la sociedad. Jugando se desarrollan capacidades y aptitudes que contribuyen a la formación de la personalidad. Por lo tanto, el juego a todas las edades es una forma de mejorar la vida social, la integración, la transmisión cultural y porqué no, una forma de aprender.

Trabajos de diferentes psicólogos destacan la importancia de la motivación y el placer en el proceso de aprendizaje de las personas [1]. El pensador Thorndike [2] declaró que recompensar a un estudiante por una acción conlleva a que quiera repetirla, si por el contrario, se le penaliza por algo que no está bien, no querrá volver a repetir en un futuro. Así, un estudiante motivado aprenderá de una forma más productiva y querrá profundizar más en el aprendizaje.

1.1. Motivación

La motivación principal por la que hacer un juego educativo es que los estudiantes aprendan y se diviertan a la vez. La idea es motivar a los alumnos para que estudien y poder facilitarles la forma de prepararse ciertas asignaturas.

Así mismo, se pretende crear un bonito juego, con interfaces fáciles de usar y con reglas entretenidas, que enganche a los jugadores con una mezcla de azar, estrategias y preguntas.

En cuanto a motivaciones personales para realizar este juego se encuentra el poder aplicar los conocimientos vistos en las clases durante la carrera, como los de programación, utilización del lenguaje de programación Java, o realización de diseño de interfaces.

1.2. Objetivos

El objetivo principal es diseñar, implementar y evaluar un programa informático que es un juego educativo. Este juego debe tener las siguientes características:

- Debe ser una aplicación de escritorio,
- Debe basarse en un tablero de bonitas interfaces,
- Se debe poder jugar con varios amigos,
- Debe servir para que los alumnos puedan aplicar los conocimientos vistos en clase, aprender cosas nuevas y divertirse al mismo tiempo.
- Deben guardarse las jugadas sin terminar para después reanudarlas.

Entre los objetivos personales de aprendizaje de este trabajo se encuentran los siguientes:

- Profundizar en la programación en Java, aplicando los conocimientos que ya adquiridos y adquiriendo nuevos con la realización de una aplicación de escritorio,
- Aprender a usar el programa Eclipse, ya que en clase no se ha usado mucho y es una herramienta fácil de utilizar y muy usada,
- Aprender algunos juegos educacionales existentes.
- Aprender conceptos nuevos de bases de datos, SQL, y MySQL, que se han visto poco durante la carrera.

En conclusión, crear un juego educativo, que mezcle la diversión y el aprendizaje, sencillo de usar y bonito, de estrategias, dinero y azar.

2. Estado del arte

Los juegos educativos siempre han estado presentes, de una forma o de otra, en la vida humana e incluso en la vida animal.

Los juegos han acompañado al hombre toda la vida, probablemente las cosquillas mezcladas con la risa sean el primer juego que existió [3]. Pero el juego no ha sido inventado por los hombres, los animales de pequeños utilizan el juego para desarrollar habilidades que les serán necesarias a lo largo de la vida, por ejemplo, comportamientos como la caza o la persecución. Paredes pintadas o tumbas egipcias hacen suponer que en la prehistoria ya se jugaba [4].

Durante la historia, se han dado innumerables definiciones de juego, pero Huizinga, (1938) [5] dejando atrás el pensamiento de su época, fijo la idea dominante en nuestro tiempo, tanto de psicólogos, pedagogos, maestros y de toda la sociedad en general: el juego es un fenómeno cultural, una actividad libre y desinteresada. Finalmente el diccionario de la Real Academia lo contempla como un ejercicio recreativo sometido a reglas en el cual se gana o se pierde [6].

Para este trabajo definiremos el juego como una actividad que se utiliza para la diversión y el disfrute de los participantes, con una serie de reglas y que en muchas ocasiones sirve como herramienta educativa.

Butler, (1983) [7] nos dio información más precisa de la efectividad del juego educativo en la enseñanza, aquí se resume:

- Los estudiantes adquieren conocimientos y habilidades.
- Se aprende más deprisa jugando que con otras metodologías, aunque la cantidad aprendida no es mucho mayor que con otros métodos.
- Resolver los problemas que presentan los juegos ayuda a la resolución de problemas en la vida real.
- La motivación para realizar actividades es mayor si hay un juego de por medio.
- Los juegos producen una mayor asistencia de los estudiantes a las clases.
- Jugar mejora la socialización.
- Los juegos hacen que se mantengan habilidades durante largo tiempo.
- La estimulación, la curiosidad y la fantasía pueden incrementar la efectividad de los juegos.
- Ayuda a alumnos de bajo rendimiento ya que muestran mayor interés y aprender habilidades y conceptos, igual o mejor que el resto.
- Los juegos en los que haya varios participantes activos son más efectivos.

- Algunos juegos son más productivos para unos estudiantes que para otros y a la inversa.
- Combinar juegos con trabajos de papel y lápiz debería ser más beneficioso.

Sin embargo las ventajas de los juegos en el ámbito educativo todavía son cuestionadas. Para muchos [8] es lo opuesto al trabajo, a la producción, o al rendimiento y esto hace que este instrumento sea marginado por el educador al pensar que el juego no es otra cosa que un merecido descanso tras el trabajo. Ortega & Lozano (1996) [9] opinan que la escuela tradicional ha marginado el juego debido a una diferencia intrínseca entre el juego y el aprendizaje, levantando una creencia falsa de rigor psicológico sobre la inutilidad de estos. De Borja (1985) [10] cree que el juego puede representar un cambio en la mentalidad y forma de actuar de escuelas que lo califican como una actividad en la que se pierde el tiempo. Estos son unos de los autores que plantean la intervención educativa a través del juego buscando un equilibro entre el juego dirigido y espontáneo que lleve hacia un aprendizaje constructivo [8].

Existen muchos juegos educativos diferentes. Algunos estos juegos educativos de ordenador se describen en los artículos [11], [12], [13] ó [14]. Cada uno de ellos tiene sus reglas y en cada uno se recompensa al estudiante de una forma diferente. A continuación se resumen los objetivos de cada juego y se comenta las diferencias más significativas con esta aplicación.

El artículo [11] habla de un juego de ordenador que simula un juego de mesa, con un dado y casillas donde se puede avanzar cuando el jugador acierta las preguntas o retroceder si responde mal. La gran diferencia del juego del artículo con *shopC* es que, *shopC* no solo se basa en responder preguntas sino también se pueden hacer estrategias comprando o manipulando dinero, así habrá más competición y más motivación entre los jugadores.

EL juego del artículo [12] convierte a los alumnos en mascotas virtuales, lo que es un planteamiento muy diferente al de la aplicación de este trabajo, que juega imitando la vida real, ir de compras a un centro comercial y usar dinero ficticio, es una manera diferente de entretener.

El juego del articulo [13] es un crucigrama desafío y el del articulo [14] es un juego de preguntas. Aunque las preguntas puedan estar en otros juegos o sistemas, shopC proporciona un conjunto de reglas diferentes de juego que no están en los artículos mencionados.

3. Diseño de la solución técnica

La aplicación desarrollada en este trabajo está separada en tres partes, una parte de la aplicación consta de las clases de la funcionalidad del juego, otra parte es la que crea la bases de datos, las tablas y las conecta al programa y la tercera parte son las clases con las interfaces de juego.

3.1. Tecnologías empleadas

Para este juego se han utilizado varias tecnologías de las que se hablará a continuación.

3.1.1. Java y Eclipse

La principal tecnología utilizada es Java, un lenguaje de programación de alto nivel orientado a objetos con el que se ha realizado la aplicación casi en su totalidad. Su multitud de librerías y clases permiten programar de una forma relativamente sencilla. La funcionalidad y la interfaz están desarrolladas con este lenguaje.

Se ha utilizado también Eclipse, un entorno de desarrollo integrado de código abierto cuyo SDK incluye herramientas para desarrollar código en Java. Además ofrece un IDE con un compilador de Java interno, que lo hace muy fácil de usar.

En la figura 3.1 se puede ver cómo se utiliza Java en Eclipse y en la sección 3.2.1 se explica más detalles de estas dos tecnologías:

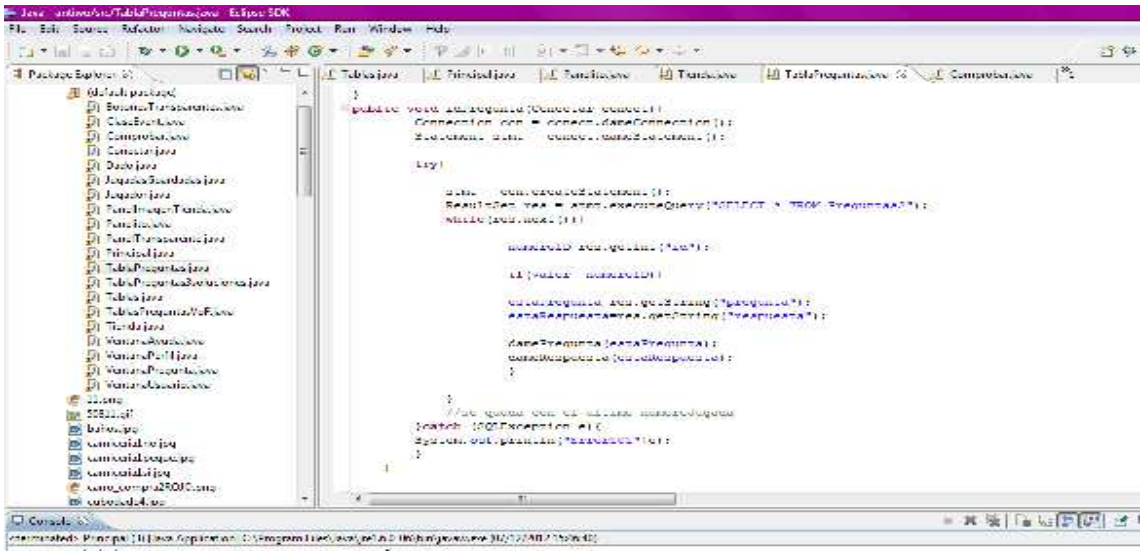


Figura 3.1: Ejemplo de una ventana de Eclipse

3.1.2. MySQL, XAMPP y SQL

Otra de las tecnologías utilizadas es MySQL que es un sistema de gestión de bases de datos relacional. Se instala en el ordenador para crear una bases de datos en donde se guardan de forma permanente cuatro tablas, una tabla para las jugadas que se van guardando para luego recuperarlas y tres tablas más con los diferentes tipos de preguntas, respuestas y las soluciones a estas. Esto está detallado en la sección 3.5 de esta memoria.

También se instaló XAMPP, que es un servidor cuyo funcionamiento es independiente de la plataforma en que se utilice, de software libre y que consta de varias partes. En este proyecto se utiliza para poder arrancar MySQL. Es muy fácil de instalar, sólo es necesario ejecutar un archivo y más fácil aún de utilizar, se arranca y para haciendo clic en un botón. En la figura 3.2 se observa como MySQL está arrancado y listo para funcionar. La figura es el panel de control que se abre del programa XAMPP.

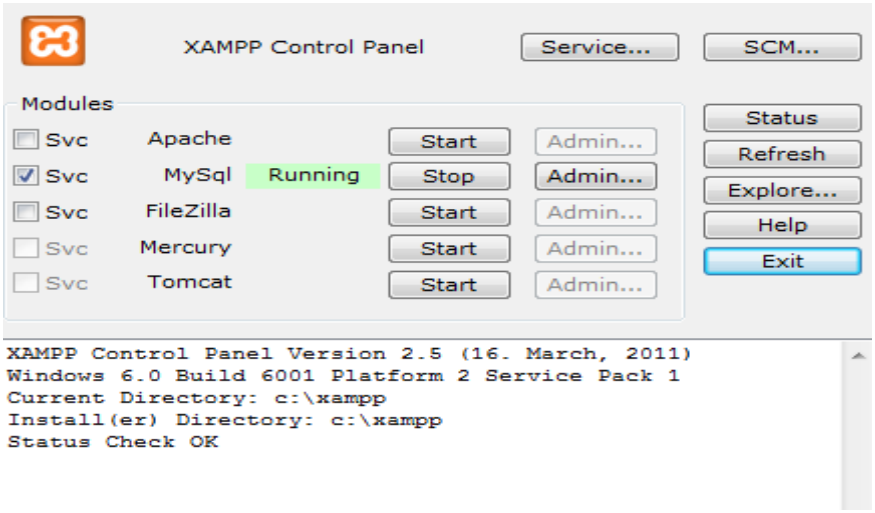


Figura 3.2. Panel de control de programa XAMPP.

Para la creación e inserción de tablas, se utiliza cómo lenguaje de programación SQL, que se conecta con la aplicación Java mediante una serie de librerías JDBC, que se explicarán con más detalle en la sección 3.2.2.2 y 3.2.2.1.

3.1.3. Otras tecnologías

En una pequeña parte de la aplicación, para la creación de una ayuda al usuario y al profesor se utilizan también los lenguajes de programación HTML y XLS. Esto se explica en la sección 3.2.4.2.

3.2.-Arquitectura

En esta sección se describen los programas y lenguajes de programación utilizados para realizar esta aplicación.

A continuación se muestra una vista general del programa en la figura 3.3:

Figura 3.3: Vista general del programa.

3.2.1. Java y Eclipse

Java es un lenguaje de programación desarrollado por James Gosling. Deriva de C y C++, está basado en clases y orientado a objeto. Es un componente de Sun Microsystems (adquirida por Oracle). Las aplicaciones en Java son compiladas a bytecode que puede correr en cualquier máquina virtual (JVM), la intención es que el programa funcione en cualquier dispositivo; Gosling, 'escríbelo una vez, ejecútalo en cualquier lugar.' Es, a partir del año 2012, uno de los lenguajes de programación más populares [15] [16].

En 2003-2004 [17] se creó la Fundación Eclipse, una entidad legal, independiente y sin ánimo de lucro que creó un producto llamado Eclipse. Eclipse es un entorno de desarrollo integrado (IDE) de código abierto y aunque tiene muchos otro usos, en esta aplicación se utiliza para desarrollar el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ). Para que Eclipse funcione correctamente en un ordenador,

hay que instalar una máquina virtual de Java (JVM), JRE (Java Runtime Environment) o JDK (Java Develper Kit) de Sun.

3.2.2.-Bases de datos

Para la creación de la base de datos se ha utilizado un gestor de bases de datos llamado MySQL y para la creación y modificación de las tablas que contiene se ha utilizado lenguaje SQL que se conecta con la aplicación Java mediante una biblioteca llamada JDBC. A continuación se describen detalladamente.

3.2.2.1. MySQL

MySQL es un sistema que gestiona las bases de datos, relacional, multiusuario y de software libre bajo licencia GNU GPL. Desde el 2008 es de Sun Microsystems (Oracle) y está escrito en C y C++. Es una base de datos muy rápida y muy buena para lectura de datos, aunque puede provocar problemas si las modificaciones en esta son muy grandes [18].

Su interfaz ODBC (Open Database Connectivity) permite a cualquier lenguaje de programación comunicarse con las bases de datos y a varios lenguajes de programación acceder a ellas.

3.2.2.2.-SQL y JDBC

Lenguaje de consulta estructurado ó SQL (Structured Query Language) es un lenguaje declarativo usado para consultar y manipular bases de datos. Se caracteriza por su facilidad para guardar información o modificarla en una base de datos [19]. En el Anexo II se explica con detalle como insertar y modificar tablas con este lenguaje.

JDBC (*Java Database Connectivity*) [20] permite manipular y consultar los datos de una base de datos desde un programa Java, es decir, es una biblioteca de Java inspirada en OCBC (permite a cualquier lenguaje de programación comunicarse con las bases de datos), su API permite mapear la clase Java con una tabla de la base de datos y facilita la persistencia de los objetos. Se trabaja con el paquete *java.sql* cuyas clases se explican en la sección 3.5 de esta memoria.

3.2.- Modelo de la lógica

Para la funcionalidad de esta aplicación se han creado cuatro clases, utilizando las clases y las librerías de Java. Estas clases se detallan en las secciones que vienen a continuación.

3.2.1-Tienda.java

El tablero del juego simula un centro comercial y cada una de las tiendas del centro comercial es una casilla del tablero. También son casillas un parking, dos baños y un centro de información. Hay tres tipos de casillas en el juego:

- **Casillas que se compran**: es donde hay una tienda o un restaurante y la mayoría son la mayoría de las casillas del tablero de juego, el bar, la carnicería, la charcutería, la droguería... El jugador que caiga en una de estas casillas podrá comprarla si no está comprada respondiendo a tres preguntas, el precio dependerá de los aciertos, como se explica en el Anexo I.
- **Casillas no comprables:** son las casillas de las esquinas del tablero y hay son de cuatro tipos:
  - **Parking:** es la primera casilla del tablero, donde se inicia el juego. Cada vez que un jugador caiga en esta casilla deberá pagar dinero.
  - **Baños:** son todas las casillas de la esquina superior derecha y de la esquina inferior izquierda del tablero y cuando cae aquí un jugador perderá dinero.
  - **Información:** es la casilla de la esquina inferior derecha. Si un jugador cae en esta casilla será premiado.

El funcionamiento de estás casillas se explica con más detalle el Anexo I.

Por la importancia de las casillas o tiendas del tablero, una de las clases principales de esta aplicación se denomina *Tienda.java* que es la encargada de caracterizar cada una de estas casillas.

Se han instanciado 32 objetos de la clase Tienda, uno por casilla. Cada objeto tiene como características:

- Un *String* con el nombre de la tienda: panadería, restaurante…
- Un *int* con el precio de cada una de las tiendas/casillas; las casillas que no se compran también son de la clase *Tienda,* y en lugar de precio al que se compra la tienda tienen guardado el dinero regalan o quitan al jugador que caiga en ellas.
- Un *array* de *boolean* con las tiendas que están compradas, *true*, o no, *false*; en el caso de las casillas que no se pueden comprar su valor en el *array* es siempre *false*, es decir, que no se pueden comprar.
- Dos *int* que hacen de coordenadas en pixeles, para saber en qué parte del tablero está situada cada tienda y luego localizar a los jugadores en el tablero.

En la figura 3.4, se presenta el tablero del juego, cada cuadro es un objeto de la clase *Tienda*.

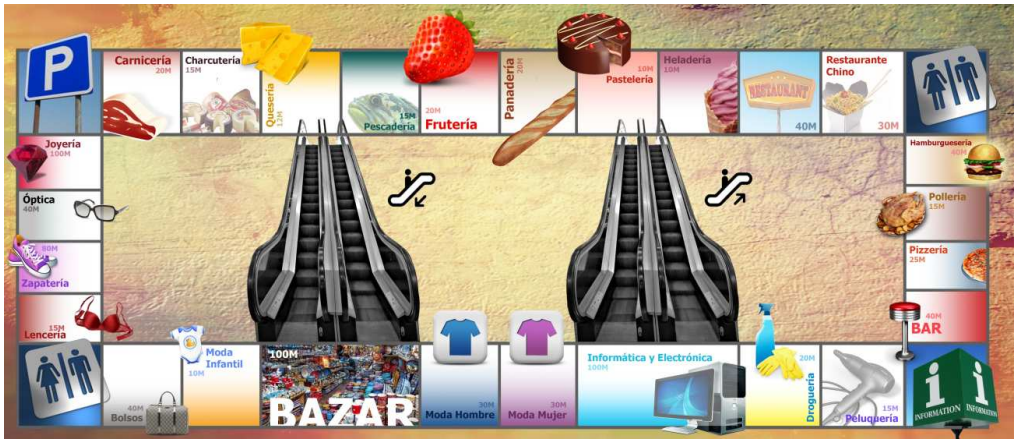


Figura 3.4. Tablero del juego.

Esta clase consta de cuatro métodos:

- ***precioTiendas(String nombreTienda):*** devuelve un *int* con el precio de una tienda determinada, que le pasamos cómo parámetro en un String .
- ***coordenadaYtienda(String nombreTienda):*** devuelve un *int* con el número coordenada de alto de una tienda determinada, que es un *String* que le pasamos cómo parámetro.
- ***coordenadaXtienda(String nombreTienda):*** devuelve un *int* con el número coordenada de ancho de una tienda determinada, que es un *String* que le pasamos cómo parámetro.
- ***comprobarCompra(String nombreTienda):*** devuelve un *array* actualizado con las casillas compradas, ya que se ha comprado la tienda "nombreTienda".

Las escaleras mecánicas, son las imágenes que unen la parte superior del tablero con la parte inferior, no son casillas y por lo tanto no son objetos de la clase *Tienda*. Su funcionamiento se detalla en el Anexo I.

3.2.2. Jugador.java

La segunda de las clases es *Jugador.java* y es la clase que caracteriza a cada uno de los jugadores. En la aplicación se instanciarán tantos objetos de esta clase cómo jugadores haya en la partida. Se puede elegir, desde un jugador hasta cuatro jugadores. Cada objeto Jugador se caracteriza por:

- Un nombre, que es un *String* y que el jugador elige al inicio del juego y con el que jugará toda la partida.

- Una imagen, que no se puede elegir y que dependiendo del número de jugador que sea será de un color u otro. Son las fichas con las que se moverá por el tablero y tiene forma de ticket, son como las imágenes de la figura 3.5. Hay cuatro tipos de fichas, una para cada jugador:

Figura 3.5. Fichas del juego.

- Un *int* llamado dinero acumulado, y que es diferente para cada uno de los jugadores conforme avanza la partida ya que es lo que el jugador usa para comprar tiendas o pagar lo que se le va indicando dependiendo en la casilla o tienda que caiga.
  El dinero acumulado se mide en M., y al principio de cada partida se reparte una cantidad inicial a cada jugador que depende del número de oponentes que tenga:
  - Un jugador: 600M.
  - Dos jugadores: 300M. cada uno.
  - Tres jugadores: 200M. cada uno.
  - Cuatro jugadores: 150M. cada uno.

Figura 3.6. Icono del dinero acumulado.

  La figura 3.6 es el icono que representa el dinero acumulador para cada jugador de la partida.
- Dos *int* con las coordenadas de la *Tienda* donde está situado el jugador en ese momento en el tablero de juego.
- El número de propiedades que ha comprado, que es un *array* de *String* dónde se guarda el nombre de las casillas compradas.

Al inicio del juego el *array* de tiendas compradas está vacío, ya que ningún jugador ha comprado ninguna tienda aún, y según el jugador vaya cayendo en las tiendas y comprándolas se va rellenando una "lista de la compra" con los nombres de las propiedades o tiendas que ha adquirido durante todo la partida.

Cada jugador tiene un panel cómo el de la figura 3.7:

Figura 3.7. Panel y componentes de un jugador.

3.2.3-Dado.java

Dado.java es otra de las clases funcionales. Esta clase simula un dado, es decir, cada vez que se llama a su única función, *tirarDado()*, esta devuelve un número aleatorio entre uno y seis. Dependiendo de este número, las fichas de los jugadores se van moviendo de una casilla/tienda a otra en el tablero.

3.2.4. Principal.java

La última clase funcional es la clase *Principal.java*. Es donde arranca la aplicación, pues es donde está el *main*. Es la clase que une la parte de la funcionalidad del juego, las interfaces y una base de datos que se crea para guardar las jugadas sin terminar para reanudarlas después y dónde se guardan también las preguntas y respuestas que se usan en la aplicación, en la sección 3.5 se explica detalladamente.

3.2.4.1.-MenuBar

Lo primero que se encuentra en esta clase es un menú como el de la figura 7.

Cuando se quiere implementar un menú horizontal en Java en la parte un *JFrame* se requiere de un objeto de la clase *JMenuBar*, que es la barra principal del menú, es horizontal y en ella van colocadas las distintas opciones. También se utilizan varios objetos de la clase *JMenu* que se añaden al *JMenuBar* y que despliegan las opciones donde se pincha para hacer cierta funcionalidad.

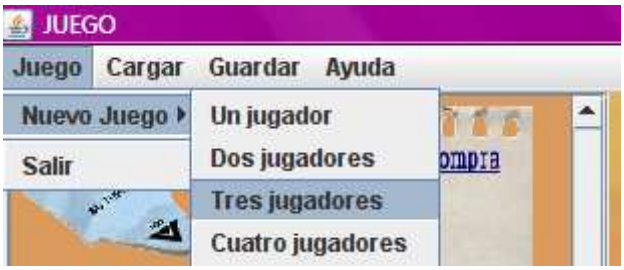


Figura 3.8. Barra menú.

La figura 3.8 presenta el menú de esta aplicación. Es un *JMenuBar* formado por cuatro *JMenu*:

- **Juego:** este *JMenu* está compuesto a su vez por dos nuevos *JMenu*.
  - *Nuevo Juego*: que se despliega en cuatro JMenuItem donde el usuario podrá elegir el número de jugadores con el que se va a jugar una nueva partida. Cuando se hace clic en uno de los JMenuItem se abre una ventana como la de la figura 8, varía en función del número de jugadores que participen en el nuevo juego. En ella, cada jugador podrá escribir el nombre con el que jugará toda la partida.

Figura 3.9. Panel para nombrar a los jugadores.

  - *Salir*: como se ve en la figura 3.9, el siguiente JMenuItem está separado por un separador del anterior y sirve para abandonar el juego en cualquier momento.
- **Cargar:** se despliega en otro *JMenuItem*, llamado *Partidas anteriores*, y sirve para cargar las partidas del juego que han sido guardadas anteriormente y poder elegir o borrar cualquiera de ellas. Esta parte se explica en la sección 3.2.4.4.
- **Guardar:** sirve para guardar en cualquier momento de la partida una jugada del juego y que no está concluida. La forma de guardarla se detalla en la sección 3.2.4.4.
- **Ayuda:** esta opción abre una ventana con ayuda para los usuarios del juego. En la sección 3.2.4.2 se explica detalladamente.

3.2.4.2-JavaHelp

Otro de los botones del menú superior es de ayuda para la instalación y utilización del juego tanto para profesores como para alumnos.

Para esto se ha utilizado una librería de Java llamada *JavaHelp*. Esta librería permite poner ventanas de ayuda para las aplicaciones.

Para poder utilizar esa librería se necesitan unos archivos jar, descargados de la web de Sun, que se importan en la aplicación en el Eclipse. Los archivos son los cuatro señalados en la figura 3.10:

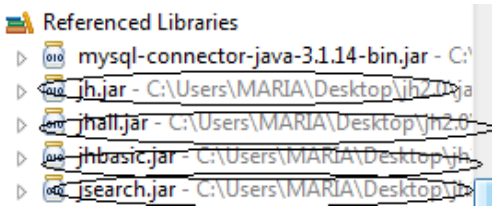


Figura 3.10. Librerías JavaHelp

En este juego la ayuda consta de una ventana que se abrirá al pulsar *Ayuda* en el menú. Es una ventana como la que se observa en la figura 3.11:

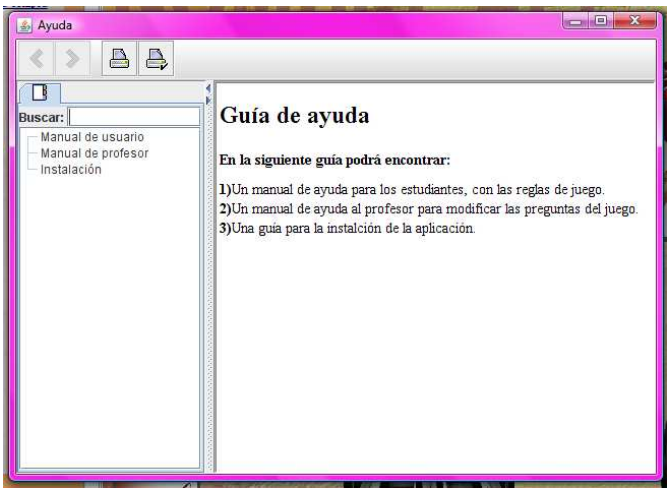


Figura 3.11. Ventana de ayuda.

Consta de varias partes:

- **Archivos .html**: Son los archivos que llevan el texto con la información. Para después identificarlo en la aplicación Java y en el resto de fichero de JavaHelp se crea un archivo con la extensión jhm, llamado fichero mapa, que no es más que un fichero xml en el que se da un nombre a cada uno de los ficheros html creados.

- **Menú de contenidos**: Se encuentra parte izquierda de la ventana. Es una tabla con los contenidos que tiene la ayuda. Este menú se declara en un fichero de formato XML, pero con extensión hs, que viene de Help Set. En este fichero se configura todo.

Para que todo funcione todo en la aplicación, se utilizan [21] los métodos *enableHelpOnButton()* y *enableHelpKey()* del *HelpBroker*. Con el método *enableHelpOnButton()*al pulsar el botón de ayuda se abre la ventana comentada y con *enableHelpKey()* al pulsar F1 la ventana de ayuda se abrirá automáticamente cuando la aplicación este abierta.

La ayuda consta de dos manuales, uno para usuarios y otros para profesores y una guía de instalación del juego en un ordenador. La propia ventana contiene dos botones para imprimir la información de ayuda, otros dos para ir para delante y para atrás y la opción de buscar. En la figura 3.12 se puede ver uno de los manuales .

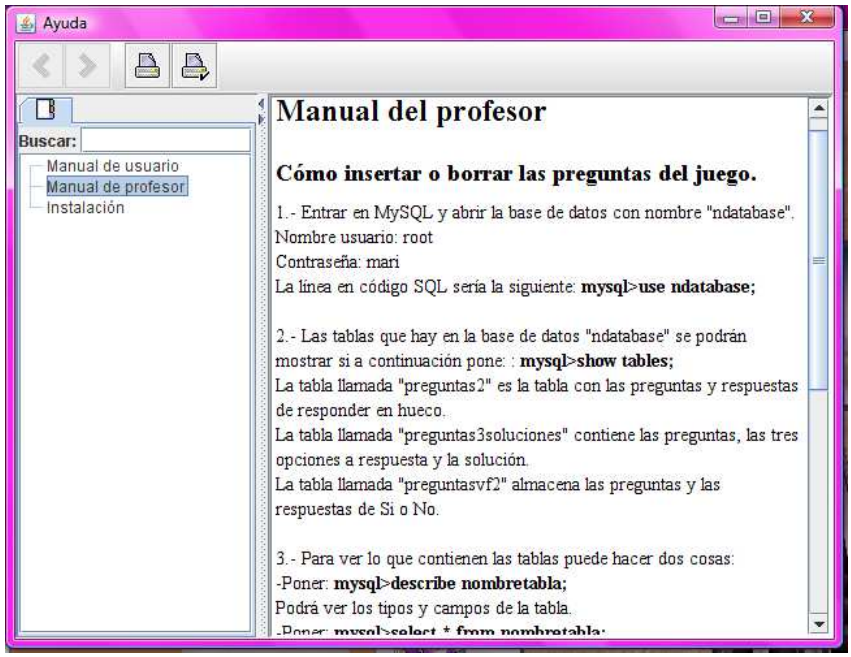


Figura 3.12. Ventana de ayuda con manual.

3.2.4.3.-Clases internas

En la clase *Principal*.java, hay varias clases internas. La primera clase que nos encontramos es *TableroFondo*, extiende de un *JPanel* y de la que darán más detalles en la sección 3.4 y cuya función será pintar el tablero, las fichas, el dado y la máquina de turnos.

La siguiente clase es *PanelSecundario*, extiende de *JPaneScroll* y es la encargada de colocar los panel *scroll* de cada jugador en la pantalla de juego, de esta clase también se darán más detalles en la sección 3.4.

Una de las clases más importante de esta aplicación es la llamada *Evento*. Esta clase implementa un *ActionListener* y contiene un método llamado *actionPerformed()*. En ella se tratan todos los eventos de ratón ocurridos en el juego, el clic en uno de los *JMenuItem* para elegir un jugador, una tirada del dado, etc.

Uno de los eventos más importante de la aplicación es "tirar el dado". Cuando un jugador hace clic sobre el dado se genera un número aleatorio del uno al seis, inmediatamente después, se buscan las coordenadas del jugador que tenga el turno para saber en qué *Tienda* del tablero se encuentra y poder repintar la ficha en el tablero en otras coordenadas, es decir, en otra tienda, dependiendo del número que haya salido en el dado.

Antes de que la ficha se repinte en el tablero, el programa comprueba si está en una casilla dónde pueda usar las escaleras mecánicas, es decir, si la ficha está en las tiendas de óptica, joyería, parking, carnicería, charcutería o quesería y el número que ha salido en el dado es tal que puede pasar por las escaleras de bajada, se le formula una pregunta como la de la figura 3.13:

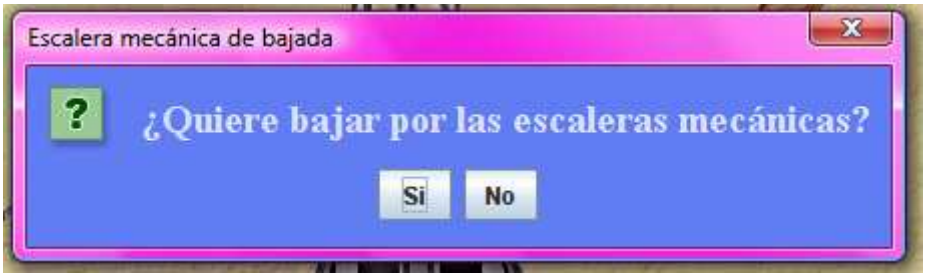


Figura 3.13. Panel de pregunta sobre las escaleras mecánicas.

Si la respuesta es *Sí*, la ficha "bajará" a la parte de debajo de tablero y dependiendo de la casilla dónde estaba anteriormente y del número de dado se colocará en una tienda u otra.

Si la respuesta es *No*, la ficha se "moverá" hacia delante en el tablero, el número de casillas correspondiente, sin tener en cuenta las escaleras.

Lo mismo ocurría en las casillas del bar, información, peluquería, droguería o informática y electrónica, pero las escaleras son de subida, por lo que te llevaría a la parte superior del tablero.

Por ejemplo, si un jugador tiene su ficha en la droguería y al tirar el dado sale un número mayor que dos, supongamos para el ejemplo que sale cuatro, el programa detecta antes de mover la ficha que antes de esas cuatro posiciones se encuentran las escaleras de subida, por lo que automáticamente, se le fórmula la siguiente pregunta: "¿Quiere subir por las escaleras mecánicas?". Si el jugador da al botón *Si*, el programa resta al número del dado las casillas que quedan hasta las escaleras, en este caso el número del dado es cuatro y desde la droguería hasta las escaleras hay una casillas por lo que la diferencia es tres. Esta diferencia es las casillas que se "moverá" la ficha a partir de la panadería, ya que es la tienda que conecta las escaleras de subida con las casillas superiores del tablero, en este caso, la ficha correspondiente se "moverá" tres tiendas a partir de las escaleras: la panadería, la heladería y el restaurante (aunque en realidad se ha movido cuatro, ya que también cuenta la tienda de informativa y electrónica). Si pulsa *No*, la ficha sigue adelante hasta llegar al bazar, moviéndose esta vez las cuatro casillas seguidas, que es el número que ha salido en el dado: informática y electrónica, moda mujer, moda hombre y bazar.

Este ejemplo se observa en la figura 3.14.

Figura 3.14. Panel explicativo sobre la funcionalidad de las escaleras mecánicas.

Una vez que se ha producido el movimiento de la ficha pueden pasar varias cosas, dependiendo de dónde se sitúe la ficha dentro del panel, es decir, dependiendo en qué casilla o tienda del panel este situada la ficha al jugador que tenga el turno le ocurrirá una cosa u otra. Existen seis tipos de casillas en el tablero de juego donde puede para el jugador:

Casillas Tienda no comprada.

Si un jugador cae en una tienda que no está comprada por ningún otro jugador de la partida, este tiene la posibilidad de comprarla y añadirla en su lista de la compra. En el caso de la quesería, por ejemplo, se formularía una pregunta como la de la figura 3.15:

Figura 3.15. panel preguntando si se quiere comprar cierta Tienda

Hay dos posibilidades, comprar o no. Si el jugador pulsa *No compro*, el turno cambiará al siguiente jugador. Si el jugador pulsa *Si compro*, se formularán tres preguntas y el precio de la tienda a comprar dependerá de los aciertos. Esta parte se detalla en el Anexo I.

Casilla Tienda comprada por otro jugador.

Si la ficha se coloca encima de una tienda que ya está comprada por otro jugador, por ejemplo si el jugador tres ha comprado con anterioridad la Tienda de de Bolsos, le aparecerá un mensaje como el de la figura 3.16:

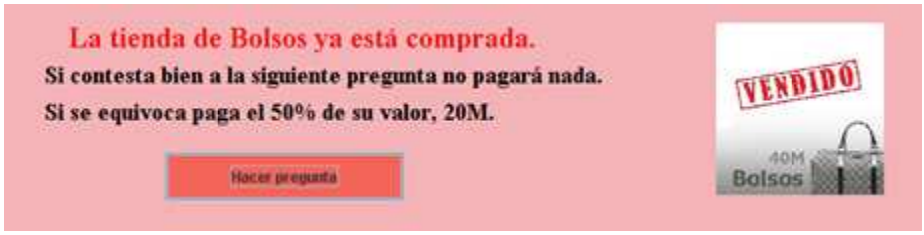


Figura 3.16. Panel mostrado al caer en una casilla comprada por otro.

Inmediatamente se le abrirá un panel con una pregunta. Si no acierta se pagará una cantidad de dinero, que es el 50% del valor de la Tienda ó nada si acierta, esta parte está detallada en esta memoria más abajo. Conteste bien o mal, el jugador perderá el turno.

Tienda comprada por el mismo jugador.

Si un jugador que ya ha comprado cierta tienda, cae en ella de nuevo, recibirá un 25% del dinero que le costó y volverá a tirar.

En la figura 3.17 aparece el panel que se abrirá si, por ejemplo, un jugador ha comprado la charcutería y vuelve a caer en ella.

Figura 3.17. Panel mostrado al caer en tienda propia.

Casilla Parking.

Cualquier jugador que caiga en esta casilla perderá 1OM. de su dinero acumulado. Se abrirá un mensaje como el de la figura 15:

Figura 3.18. Panel informativo del Parking.

Casillas Baños

Lo que ocurre en los Baños depende de si hay un jugador en la partida o más de un jugador:

- Un jugador:

Se abrirá un panel como el de la figura 3.19 y pagará 5M. de su dinero acumulado.

Figura 3.19. Panel informativo de los baños de un jugador.

- Varios jugadores:

Si en la partida hay más de un jugador, el jugador que pare en cualquiera de los dos baños recibirá un mensaje como el de la figura 3.20:

Figura 3.20. Panel informativo de los baños.

El jugador podrá elegir si pagar 5M. de su dinero acumulado y perder el turno ó la segunda opción será pagar 10M. y saltar el turno al jugador que le precede.

Casilla Información

Si cualquier jugador cae en esta casilla recibe un cheque de 10M. que se sumará a su dinero acumulado y pierde el turno. Un panel como el de la figura 3.21 será lo que informe a dicho jugador del cheque.

Figura 3.21. Panel informativo de la casilla Información.

El resto de clases internas de la clase Principal, tienen que ver con las preguntas. Hay dos tipos de preguntas, y por lo tanto dos clases:

- Preguntas de comprar: este tipo de preguntas contiene a su vez, tres tipos:
  - Preguntas de rellenar.
  - Preguntas con tres opciones.
  - Preguntas de verdadero o falso.

Son las preguntas que salen cuando el jugador quiere comprar una tienda. Cuando un jugador cae en una tienda que no está comprada y la quiere comprar se le formulan

tres preguntas, una de cada tipo, que tendrá que responder para saber el precio final de la Tienda, que dependerá de las respuestas correctas a las preguntas.

Estás preguntas están impresas en un JFrame cómo JLabel. En la figura 3.22 se puede observar la colocación de las preguntas y las respuestas.

La primera es de tipo a., la segunda de tipo b., para contestar hay que pulsar el botón de los tres cuya respuesta se crea correcta y la tercera, tipo c., es la de verdadero o falso, donde hay dos CheckBox y hay que pulsar la cajita del que se piense correcto.

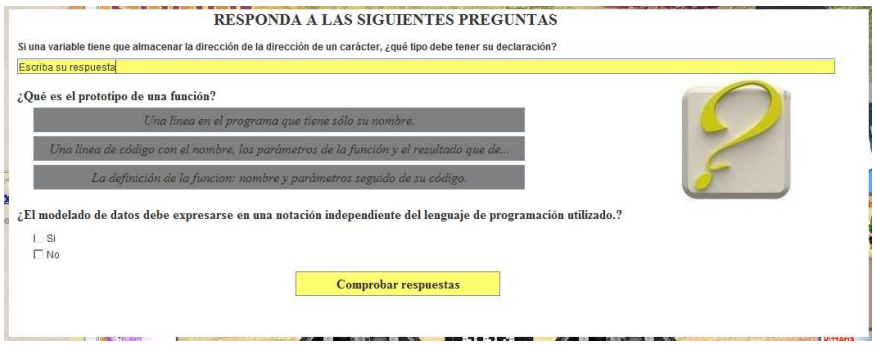


Figura 3.22. Panel con preguntas.

Inmediatamente después de contestar se comprueban las respuestas (haciendo clic en el botón inferior, “Comprobar respuestas” que se ve en la figura 18). Se abre un nuevo *JFrame* indicando las soluciones correctas (figura 3.23 solución incorrecta y figura 3.24 soluciones correctas). Si el jugador no acierta ninguna, paga el 100% del valor de la tienda. Si acierta una pagará un 75%, si acierta dos un 50% y si acierta las tres solo un 25% del valor inicial de la tienda; precio que se le restará de su dinero acumulado. Pase lo que pase, el turno cambia al siguiente jugador.

Estas preguntas están almacenadas de forma persistente en una base de datos, y son extraídas aleatoriamente. En la sección 3.2.4.4 se explica esta parte detalladamente.

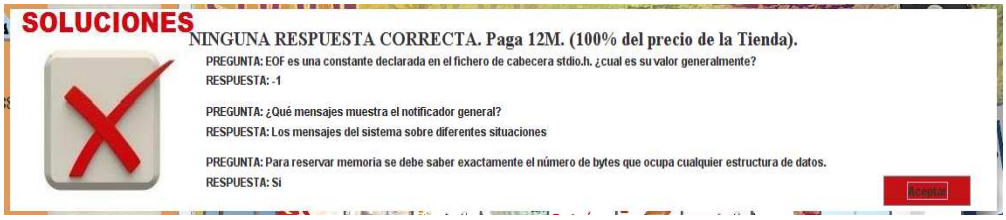


Figura 3.23. Panel de solución o soluciones incorrectas.

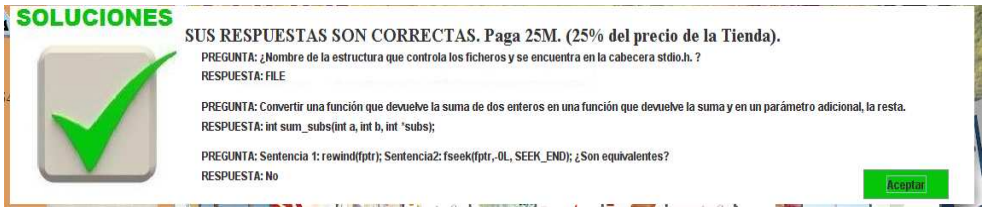


Figura 3.24. Panel de soluciones correctas.

- Preguntas de Tiendas compradas por otro jugador: cuando un jugador cae en una Tienda que ya está comprada por otro jugador, también se le formula una pregunta, como la de la figura 3.25, con una pregunta aleatoria que deberá contestar para no perder dinero. Si no acierta paga un 50% del precio inicial de la Tienda y si no, se pasa el turno al siguiente jugador.

Figura 3.25: Pregunta formulada al caer en una casilla comprada por otro jugador.

Todas las preguntas y respuestas están sacadas de la web de la asignatura de Arquitectura de Sistemas de las Universidad Carlos III.

http://www.it.uc3m.es/labas/course_notes/data_types_es.html

Para comprobar qué jugador es el ganador y perdedor de juego se hacen dos cosas:

- Comprobar si el dinero acumulado de un jugador es negativo cada vez que algún jugador modifica cantidad, si es negativo el juego termina.
  El ganador será aquel que tenga más dinero acumulado, sin sumar el coste de las propiedades que tiene adquiridas.
  El perdedor será el jugador que haya dejado su cantidad de dinero acumulado en negativo.
- Cada vez que se cambia el turno de un jugador a otro se comprueba si ha habido 50 turno, si los ha habido, el juego termina.

El jugador que tenga más dinero acumulado, en el momento de que se llegue a los 50 turnos, será el ganador del juego y el perdedor el que menos dinero acumulado sume en su cuenta.

3.2.4.4.-Guardar y cargar jugadas

Si en cierto momento, no se puede acabar una partida, pero los jugadores quieren continuarla en otro momento, esta aplicación permite guardar jugadas para después seguir jugando donde se dejo. Para ello en el menú superior de la ventana principal de juego están las opciones de *Cargar* y *Guardar*.

El botón *Guardar* da la opción de guardas la partida actual. Cuando la partida se guarda aparecerá un mensaje informativo cómo el de la figura 3.26:

Figura 3.26. Pregunta formulada al caer en una casilla comprado por otro jugador.

Solo se pueden guardar cuatro partidas, si se intentan guardar más aparecerá un mensaje como el de la figura 3.27 y la jugada no se guardará:

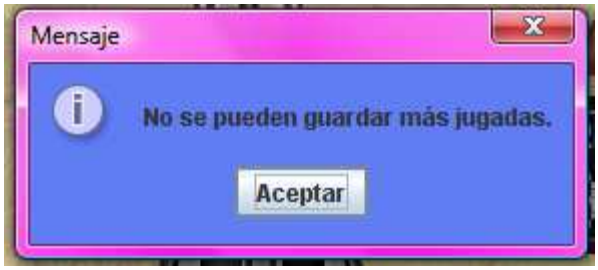


Figura 3.27. Panel informativo de que no se pueden guardar más jugadas.

Mediante el botón *Cargar* del menú, se puede recuperar una jugada que ha sido guardado anteriormente. En la opción *Partidas anteriores* se abre una ventana como la de la figura 3.28:

Figura 3.28. Panel para buscar las jugadas guardadas.

En la parte izquierda de la ventana están los juegos guardados, con una breve información sobre ellos, el número de jugadores, el nombre de los jugadores y el dinero acumulado que tenía cada uno en el momento en que se guardo la jugada. Si se quiere recuperar uno juego se pulsa encima de *JUEGO* y se abrirá una ventana con varias opciones. Una opción es *Borrar* el juego guardado, es decir, eliminarlo de la lista de jugadas anteriores, si se pulsa aquí, no se podrá volver a recuperar, ya que también se borrará de la base de datos; la otra opción es *Jugar*, en este caso, se cargará en la ventana el juego guardado para jugar.

Si por ejemplo algún jugador desea borrar la jugada guardada número dos, tendría que hacer clic en *JUEGO 2* y se le abría una ventana como la de la figura 3.29. El usuario tendría que dar a borrar y automáticamente el juego se eliminaría de la lista y de la base de datos.

Figura 3.29. Opciones que hay con una jugada guardada.

La forma de guardar toda la información de los juegos es mediante tablas en una base de datos MySQL, esto está explicado con todo detalle en el apartado 3.5. de esta memoria.

3.3. Modelo de interfaz gráfico

La parte gráfica la forman sobretodo paneles, botones y etiquetas, principalmente.

Nada más abrir la aplicación se abre el JFrame principal que contiene un gran panel (JPanel) que a la vez contiene otras dos clases de paneles:

- **Panel de Fondo** (clase: *TableroFondo* (interna en *Principal.java*)): es el panel principal de la aplicación. Consta de la imagen del tablero de fondo del juego (figura 3), de un dado, del panel con la imagen de la máquina de turnos. Es el panel donde se van pintando las imágenes de las fichas, es decir, donde se van "moviendo" las fichas de los jugadores. También consta de una matriz de botones transparentes colocados "encima" de cada casilla del tablero para dar información de las casillas a los jugadores.
- **Panel Secundario** (clase: *PanelSecundario* (interna en *Principal.java*)): es el panel lateral izquierdo, donde está la información de los jugadores. Dependiendo del número de jugadores que participen en el juego, habrá dentro de este uno, dos, tres o cuatro *JScrollPane* independientes, uno para cada jugador, que constan de imágenes y etiquetas.

En la figura 3.30 se puede observar, en la imagen de la izquierda un panel para dos jugadores y en la imagen de la derecha para cuatro jugadores.

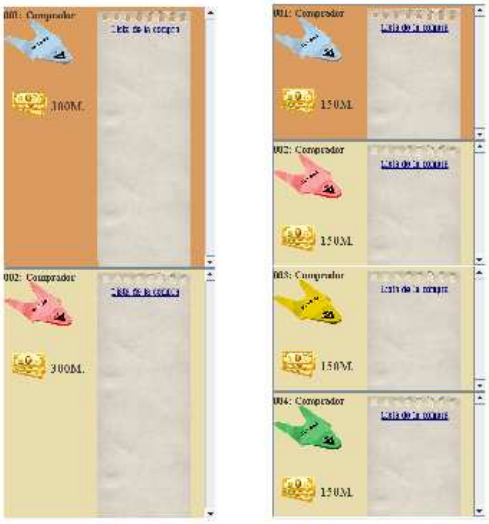


Figura 3.30. Panel de varios jugadores.

- **Paneles Transparente con imágenes** (clase: *Panelito.java*): es una clase extendida de un JPanel. Dependiendo del parámetro que con que se le cree, dibuja en un JPanel transparente una imagen u otra. Con esta clase se crean muchas de las imágenes de la aplicación y se colocan en cualquier parte de los *JPanel*. Los paneles en los se utiliza esta clase son: Las imágenes de los paneles para poner el nombre de los jugadores, el panel con icono de la interrogación al hacer las preguntas, soluciones incorrectas y correctas, las tarjetas informativas de cada tienda y las imágenes en los paneles Secundarios (ficha, billete y nota) y algunos paneles informativos. En la figura 3.31 se ven algunos de los paneles creados con esta clase:

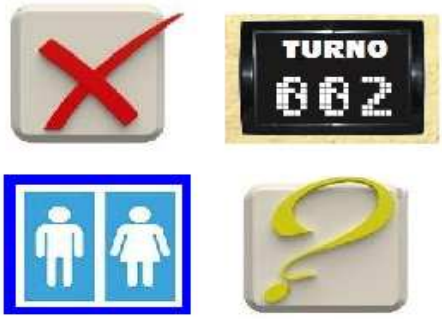


Figura 3.31. Paneles e imágenes usados.

Sobre el tablero de fondo se han colocado botones transparentes, son otra clase:

- **Botones transparentes** (clase: *BotonesTransparentes.java*), extiende de la clase *JButton* con la característica de que son transparentes para que el usuario no los vea. Están colocados encima del tablero en cada una de las casillas, en las escaleras y en el dado:
  - **Tiendas:** son los *JFrames* que se abren al hacer clic en cualquiera de las tiendas, sirven para dar más información a los jugadores. Si la tienda no está comprado la informan del precio y si está vendida así lo indican. Por ejemplo el caso de la carnicería sin comprar, y la carnicería comprada de la figura 3.32:

Figura 3.32. Panel Tienda comprada y vendida.

- **Resto casillas**: son *JFrames* que se abren al hacer clic en las casillas de parking, información, baños o escaleras. Dan información sobre lo que ocurre al caer en determinada casillas. En la figura 3.33 se ven las de las escaleras.

Figura 3.33. Panel informativo sobre las escaleras mecánicas.

- **Dado:** es un botón transparente con una imagen que va cambiando según el número aleatorio, del uno al seis, haya devuelto la clase *Dado* a la que llama al pulsar el botón. En la figura 3.34 se muestran las imágenes que forman el dado.

Figura 3.34. Dado.

El juego podrá acabar por dos motivos, cuando un jugador se queda sin dinero acumulado, con menos de 0M. ó cuando han pasado 50 turnos durante una partida.

Si un jugador se queda sin dinero perderá automáticamente y el jugador ganador será el que más dinero tenga en su dinero acumulado, sin sumar la cuantía de sus propiedades.

Si el juego acabase porque ha habido más de 50 turnos, el ganador será el que más dinero acumulado posea en ese momento, sin contar las propiedades que posea; por el contrario habrá perdido el jugador que menos cantidad de dinero acumulado tenga a su disposición.

Cuando el juego finaliza se hace un resumen de la situación de cada jugador y se da la opción de comenzar a jugar de nuevo o de salir del juego. En la figura 3.35 se puede observar un ejemplo:

Figura 3.35. Panel final del juego.

En el ejemplo de la figura 31, el jugador 001, que es Julia, tiene -75M., por lo que es la perdedora y Enrique, el jugador número 003, ha ganado ya que se ha quedado con más dinero que el resto cuando Julia se arruinó.

Al final el juego y el recuento la aplicación da dos opciones, volver a jugar un nuevo juego o salir del juego.

Todos los botones, paneles o imágenes están colocados dependiendo de los pixeles de la pantalla. Cuando arranca la aplicación se calcula la resolución de la pantalla en la que se está arrancando y mediante porcentajes se van calculando los pixeles adecuados.

Para la realización de las imágenes, tablero, fichas, etc. se han utilizado programas como el Paint, Photoshop ó GIMP 2.

Las imágenes que se ven en la figura 3.36, están elaboradas completamente, el tablero, las tarjetas de información de las casillas, la máquina de los turnos y los paneles de las soluciones a las preguntas.

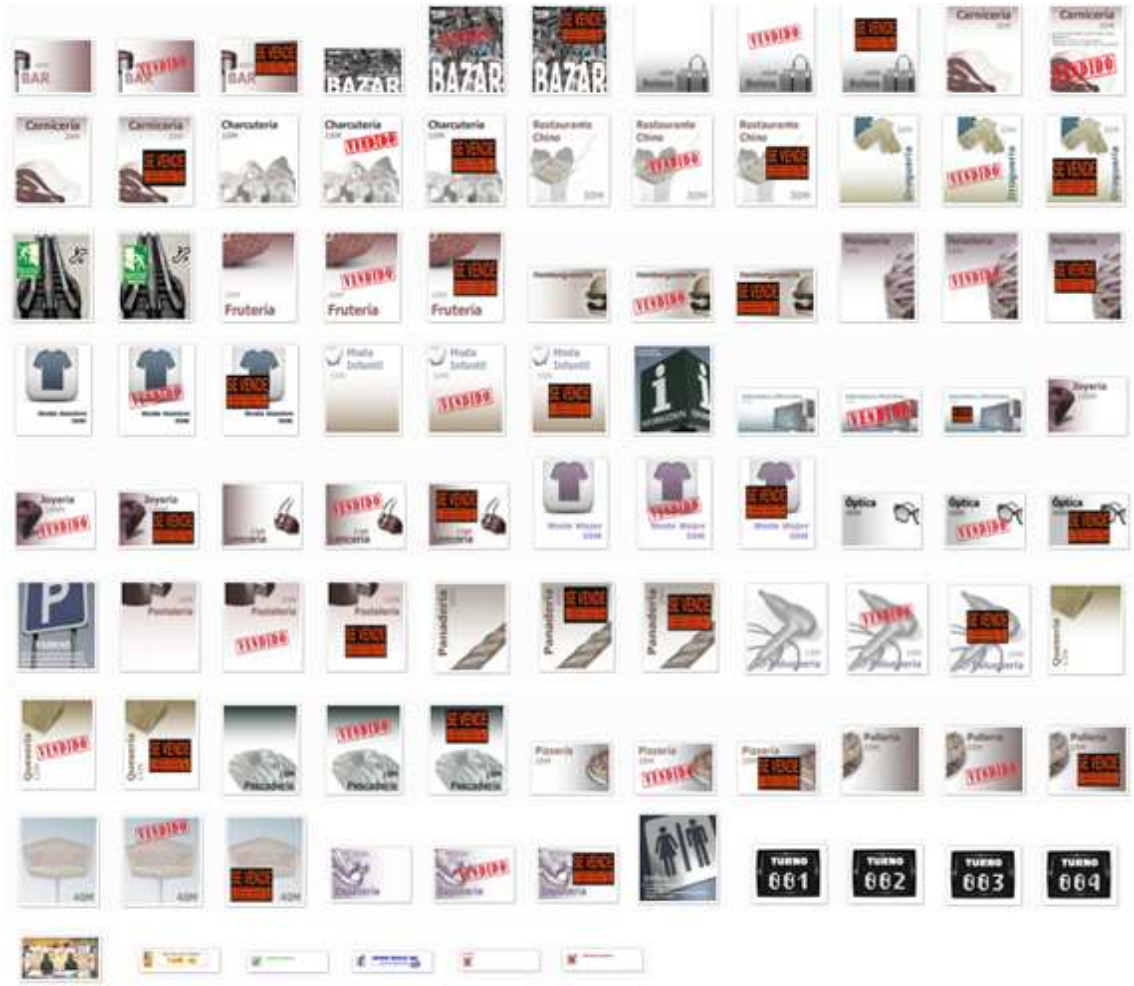


Figura 3.36.Imagenes elaboradas

Las figura 3.37 muestra las imágenes modificadas:

Figura 3.37. Imágenes modificadas

3.5.-Base de datos

Para crear una base de datos para la aplicación se ha utilizado MySQL y para manipularla JDBC, que como se ha explicado con anterioridad, permite modificar y consultar los datos desde un programa Java.

Se han utilizado las siguientes clases del paquete java.sql:

- *DriverManager*: Para cargar el controlador del conector
- *Connection:* Permite establecer conexiones con la base de datos
- *Statement*: Permite ejecutar sentencias SQL y enviarlas a la BBDD
- *ResultSet*: Para almacenar el resultado de la consulta

Con la aplicación XAMP sea arranca MySQL y en el programa Java se crea una base de datos. Para establecer la conexión entre la base de datos y la aplicación Java se carga el controlador del conector con la clase *DriverMananger*, el conector se importa en Eclipse en un archivo *.jar*, véase la figura 3.38, y después se conecta MySQL con la aplicación mediante la clase *Connection*, con el usuario y contraseña de MySQL.

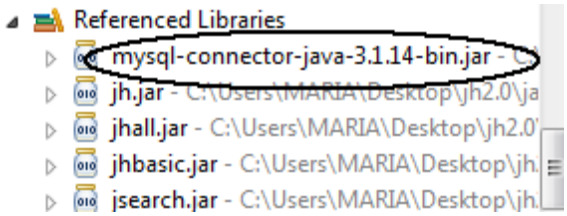


Figura 3.38. Archivo importado de formato jar para el funcionamiento de MySQL.

Se utilizan sentencias SQL como *create table* para crear las tablas en la base de datos ó *INSERT TO* para insertar el contenido de las tablas. Para extraer contenido de las tablas se utiliza *ResultSet*, que me almacena el su contenido.

La base de datos que se utiliza se llama *test* y se crea por defecto al instalar MySQL. Para utilizarla solo hay que poner el comando *use test* en la línea de comandos. En la figura 3.39 se muestra en el intérprete de línea de comandos las cuatro tablas que contiene la base de datos *test*.

```
mysql> use test;
Database changed
mysql> show tables;
+-----------------------+
| Tables_in_test        |
+-----------------------+
| jugadores             |
| preguntas2            |
| preguntas3soluciones2 |
| preguntasvf2          |
+-----------------------+
4 rows in set (0.29 sec)
```

Figura 3.39. Cómo abrir la base de datos y ver las tablas con SQL.

A continuación se detalla el contendido de cada una de las tablas:

- **Tabla jugadas guardadas**: esta tabla contiene las partidas guardadas con anterioridad.

Este tipo de tabla es de tamaño variable, es decir, dependiendo de los jugadores de la partida, a la tabla se añadirán entre una y cuatro filas, una por jugador.

Las columnas son siempre iguales, en la primera columna se guarda el número de jugada, sólo se pueden guardar cuatro jugadas, es de tipo *int* y sirve para recuperar después una jugada en concreto, es decir, si hay en una tabla tres jugadores con un mismo número de jugada es que los tres son de una misma partida guardada anteriormente.

En la segunda columna se almacena el nombre con el que el jugador se identificó al principio del juego, es de tipo *varchar*.

En la tercera columna se guarda en un *integer*, el número del jugador que tenía el turno en el momento de guardar la partida, para que al recuperarla, ese jugador sea el que inicie el juego.

En la cuarta columna los jugadores de cada partida con tipo *varchar,* para localizar después qué datos corresponden a cada jugador.

En la quinta y sexta columna se guardan las coordenadas del tablero donde se situaba el jugador en el momento que se guardó la jugada, son de tipo *int*eger.

La columna siete almacena el dinero acumulado de cada jugador, también en el momento que se guarda el juego, es de tipo int.

Las restantes 29 columnas siguientes, son todas de tipo *varchar*, y es donde se almacena el nombre de las Tiendas que están compradas por cada jugador. En figura 3.40 se observa un ejemplo de este tipo de tablas.

numeroJugada	reNombreJugador	turnoGuardado	nombreJugador	posicionX	posicionY	dineroAcumulado	Tienda0	Tienda1
1	Rosa	1	Jugador1	404	600	260		
1	Pepe	1	Jugador2	640	168	285		
2	Raul	3	Jugador1	768	168	200		
2	Pedro	3	Jugador2	404	168	180		CarnicerÝa
2	Toni	3	Jugador3	318	175	200		

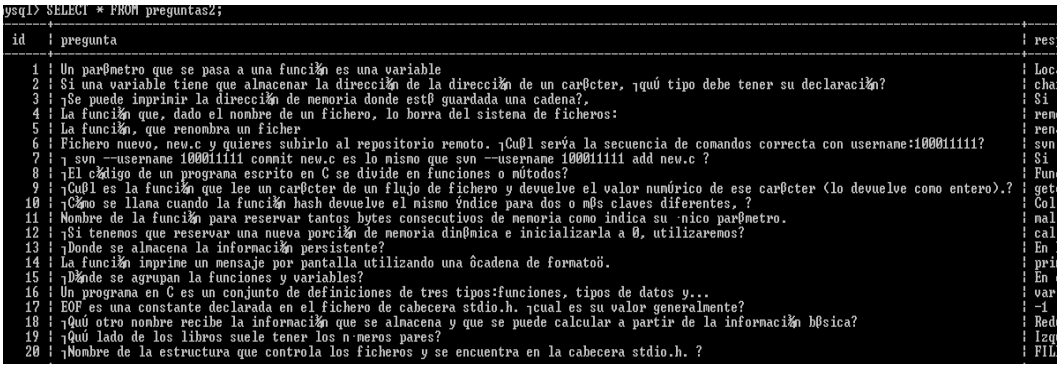
Figura 3.40. Tabla de las jugadas guardadas.

En la figura 3.40, se puede ver una parte de la tabla llamada *jugadores*. Esta tabla guarda dos jugadas, una de dos jugadores y otra de tres jugadores. En la segunda partida, el jugador dos, llamado Pedro, ha comprado la charcutería, que se almacena

en la tabla para después recuperarlo junto al resto de información. En esta figura la tabla no está completa ya que tiene 36 columnas y sólo se ven las primeras.

Si un jugador borrar una jugada guardada, esta se borrará automáticamente de la tabla llamada jugadores donde estaba almacenada.

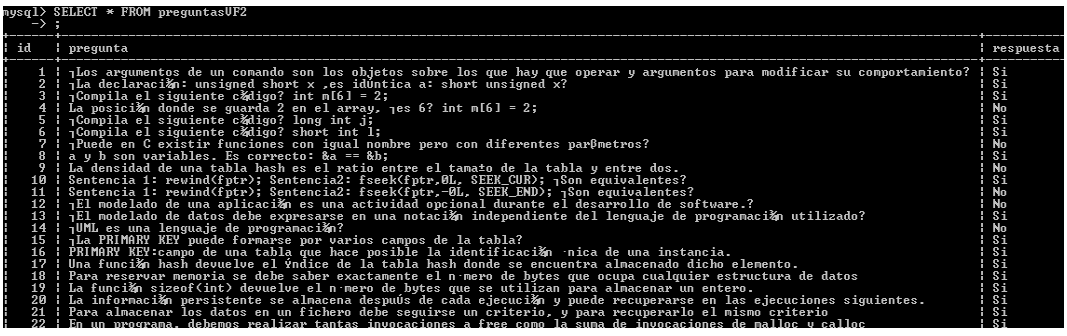
- **Tabla preguntas rellenar:** una tabla como la de la figura 3.41, es la que almacena las preguntas y respuestas en las que hay que contestar con una palabra o frase.
  La primera contiene un número identificador, para coger una aleatoriamente y localizarlas bien, la segunda columna guarda las preguntas y la segunda las respuestas.



```
ysql> SELECT * FROM preguntas2;
+----+-----------------------------------------------------------------------------------------------------------------+-----+
| id | pregunta                                                                                                        | res |
+----+-----------------------------------------------------------------------------------------------------------------+-----+
|  1 | Un par0metro que se pasa a una funci%n es una variable                                                          | Loc |
|  2 | Si una variable tiene que almacenar la direcci%n de la direcci%n de un car0cter, ¬qué tipo debe tener su declarac%n? | cha |
|  3 | ¬Se puede imprimir la direcci%n de memoria donde est0 guardada una cadena?,                                     | Si  |
|  4 | La funci%n que, dado el nombre de un fichero, lo borra del sistema de ficheros:                                 | rem |
|  5 | La funci%n, que renombra un ficher                                                                              | ren |
|  6 | Fichero nuevo, new.c y quieres subirlo al repositorio remoto. ¬Cu0l ser%a la secuencia de comandos correcta con username:100011111? | svn |
|  7 | ¬ svn --username 100011111 commit new.c es lo mismo que svn --username 100011111 add new.c ?                    | Si  |
|  8 | ¬El c%digo de un programa escrito en C se divide en funciones o m0todos?                                        | Fun |
|  9 | ¬Cu0l es la funci%n que lee un car0cter de un flujo de fichero y devuelve el valor numúrico de ese car0cter (lo devuelve como entero).? | get |
| 10 | ¬C%mo se llama cuando la funci%n hash devuelve el mismo ýndice para dos o m0s claves diferentes, ?              | Col |
| 11 | Nombre de la funci%n para reservar tantos bytes consecutivos de memoria como indica su -nico par0metro.         | mal |
| 12 | ¬Si tenemos que reservar una nueva porci%n de memoria din0mica e inicializarla a 0, utilizaremos?              | cal |
| 13 | ¬Donde se almacena la informaci%n persistente?                                                                  | En  |
| 14 | La funci%n imprime un mensaje por pantalla utilizando una ôcadena de formatoö.                                  | pri |
| 15 | ¬D%nde se agrupan la funciones y variables?                                                                     | En  |
| 16 | Un programa en C es un conjunto de definiciones de tres tipos:funciones, tipos de datos y...                    | var |
| 17 | EOF es una constante declarada en el fichero de cabecera stdio.h. ¬cual es su valor generalmente?               | -1  |
| 18 | ¬Qué otro nombre recibe la informaci%n que se almacena y que se puede calcular a partir de la informaci%n b0sica? | Red |
| 19 | ¬Qué lado de los libros suele tener los n-meros pares?                                                          | Izq |
| 20 | ¬Nombre de la estructura que controla los ficheros y se encuentra en la cabecera stdio.h. ?                     | FIL |
+----+-----------------------------------------------------------------------------------------------------------------+-----+
```

Figura 3.41. Tabla de preguntas con una solución.

- **Tabla preguntas de si o no:** en una tabla como la de la figura 3.42, se guarda un identificador en la primera columna, en la segunda columna las preguntas, y en otro la respuesta correcta, que siempre será “Si” o “No”.

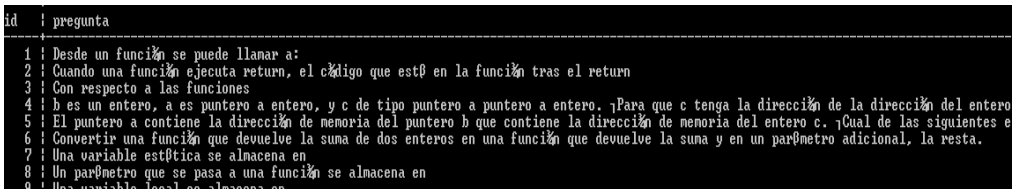


```
mysql> SELECT * FROM preguntasUF2
    -> ;
+----+-----------------------------------------------------------------------------------------------------------------+-----------+
| id | pregunta                                                                                                        | respuesta |
+----+-----------------------------------------------------------------------------------------------------------------+-----------+
|  1 | ¬Los argumentos de un comando son los objetos sobre los que hay que operar y argumentos para modificar su comportamiento? | Si |
|  2 | ¬La declarac%n: unsigned short x ,es id0ntica a: short unsigned x?                                              | Si        |
|  3 | ¬Compila el siguiente c%digo? int m[6] = 2;                                                                     | Si        |
|  4 | La posici%n donde se guarda 2 en el array, ¬es 6? int m[6] = 2;                                                 | No        |
|  5 | ¬Compila el siguiente c%digo? long int j;                                                                       | Si        |
|  6 | ¬Compila el siguiente c%digo? short int 1;                                                                      | Si        |
|  7 | ¬Puede en C existir funciones con igual nombre pero con diferentes par0metros?                                  | No        |
|  8 | a y b son variables. Es correcto: &a == &b;                                                                     | Si        |
|  9 | La densidad de una tabla hash es el ratio entre el tamato de la tabla y entre dos.                              | No        |
| 10 | Sentencia 1: rewind(fptr); Sentencia2: fseek(fptr,0L, SEEK_CUR); ¬Son equivalentes?                             | Si        |
| 11 | Sentencia 1: rewind(fptr); Sentencia2: fseek(fptr,-0L, SEEK_END); ¬Son equivalentes?                            | No        |
| 12 | ¬El modelado de una aplicaci%n es una actividad opcional durante el desarrollo de software.?                   | No        |
| 13 | ¬El modelado de datos debe expresarse en una notaci%n independiente del lenguaje de programaci%n utilizado?     | Si        |
| 14 | ¬UML es una lenguaje de programaci%n?                                                                           | No        |
| 15 | ¬La PRIMARY KEY puede formarse por varios campos de la tabla?                                                   | Si        |
| 16 | PRIMARY KEY:campo de una tabla que hace posible la identificaci%n -nica de una instancia.                       | Si        |
| 17 | Una funci%n hash devuelve el ýndice de la tabla hash donde se encuentra almacenado dicho elemento.              | Si        |
| 18 | Para reservar memoria se debe saber exactamente el n-mero de bytes que ocupa cualquier estructura de datos      | Si        |
| 19 | La funci%n sizeof(int) devuelve el n-mero de bytes que se utilizan para almacenar un entero.                    | Si        |
| 20 | La informaci%n persistente se almacena despuús de cada ejecuci%n y puede recuperarse en las ejecuciones siguientes. | Si |
| 21 | Para almacenar los datos en un fichero debe seguirse un criterio, y para recuperarlo el mismo criterio         | Si        |
| 22 | En un programa, debemos realizar tantas invocaciones a free como la suma de invocaciones de malloc y calloc     | Si        |
+----+-----------------------------------------------------------------------------------------------------------------+-----------+
```

Figura 3.42. Tabla de preguntas sí o no.

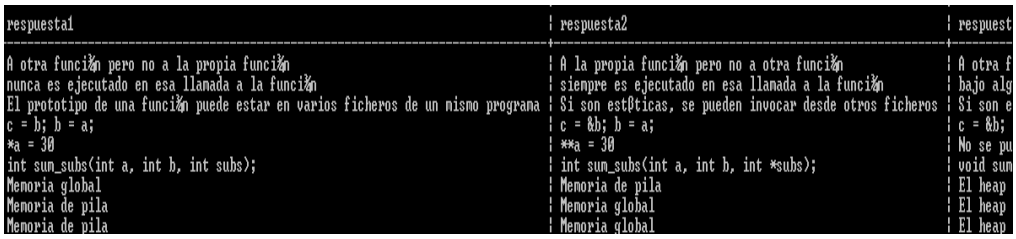
- **Tabla preguntas multirespuesta:** La figuras 3.43, 3.44, 3.45 es una tabla que contiene cuatro columnas, la primera guarda un identificador, la segunda almacena la pregunta, las tres siguientes con las tres respuestas posibles y una cuarta con la solución correcta. Las siguientes imágenes son partes de una tabla, la primera

figura muestra la primera columna con los id y la segunda con las preguntas; en la siguiente se ven las tres respuestas posibles para cada preguntas y en la última fila de la tabla esta la solución. En las siguiente figuras se muestra una tabla completa, la figura 3.43 es de las dos primeras columnas, el identificador de la pregunta y la pregunta en sí; la figura 3.44 muestra las tres posibles respuestas y la figura 3.45 guarda la solución correcta a la preguntas, para luego hacer la comprobación.



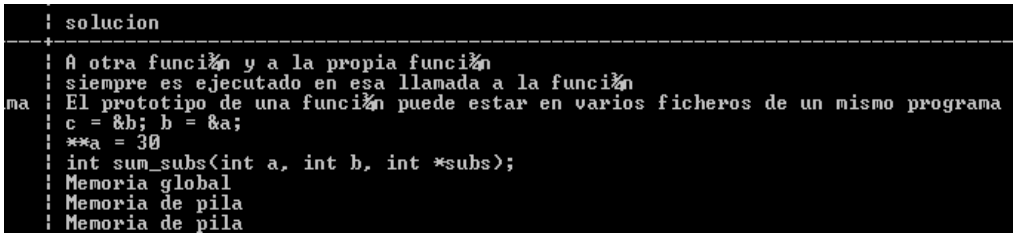
id	pregunta
1	Desde un funci¾n se puede llamar a:
2	Cuando una funci¾n ejecuta return, el c¾digo que est en la funci¾n tras el return
3	Con respecto a las funciones
4	b es un entero, a es puntero a entero, y c de tipo puntero a puntero a entero. ¿Para que c tenga la direcci¾n de la direcci¾n del entero
5	El puntero a contiene la direcci¾n de memoria del puntero b que contiene la direcci¾n de memoria del entero c. ¿Cual de las siguientes e
6	Convertir una funci¾n que devuelve la suma de dos enteros en una funci¾n que devuelve la suma y en un par metro adicional, la resta.
7	Una variable est tica se almacena en
8	Un par metro que se pasa a una funci¾n se almacena en

Figura 3.43. Primera y segunda columna de la tabla de preguntas multirespuesta.



respuesta1	respuesta2	respuest
A otra funci¾n pero no a la propia funci¾n	A la propia funci¾n pero no a otra funci¾n	A otra f
nunca es ejecutado en esa llamada a la funci¾n	siempre es ejecutado en esa llamada a la funci¾n	bajo alg
El prototipo de una funci¾n puede estar en varios ficheros de un mismo programa	Si son est ticas, se pueden invocar desde otros ficheros	Si son e
c = b; b = a;	c = &b; b = a;	c = &b;
*a = 30	**a = 30	No se pu
int sum_subs(int a, int b, int subs);	int sum_subs(int a, int b, int *subs);	void sum
Memoria global	Memoria de pila	El heap
Memoria de pila	Memoria global	El heap
Memoria de pila	Memoria global	El heap

Figura 3.44. Columnas de posibles soluciones de preguntas-multirespuesta.



	solucion
	A otra funci¾n y a la propia funci¾n
	siempre es ejecutado en esa llamada a la funci¾n
ma	El prototipo de una funci¾n puede estar en varios ficheros de un mismo programa
	c = &b; b = &a;
	**a = 30
	int sum_subs(int a, int b, int *subs);
	Memoria global
	Memoria de pila
	Memoria de pila

Figura 3.45. Columna de la tabla de preguntas-multirespuesta con la solución.

Insertar o eliminar preguntas en las tablas se hace por línea de comandos utilizando código SQL, esto se explica en el Anexo 2.

4.-Evaluación y resultados

4.1.-Evaluación

Para evaluar esta aplicación se han aplicado métodos cuantitativos, realizando unas encuestas a diez compañeros de la Universidad Carlos III de Madrid.

Los compañeros han realizado dos tipos de encuestas, la primera de ellas antes de probar el juego y la segunda después de jugar. Para diseñar las preguntas de estas encuestas, se ha tomado como base y adaptado un cuestionario utilizado en los años 2011 y 2012 en la asignatura de Laboratorio de Arquitectura de Ordenadores para la evaluación de un software de competición.

Con esto se pretende saber un poco más sobre que piensan los alumnos sobre este juego educativo y saber si la aplicación realizada les hace disfrutar y aprender, que es el objetivo final de este proyecto.

Con la encuesta antes de jugar, se pretende saber si al alumno le gustan en general los juegos para la educación y si es competitivo de una forma negativa. La encuesta ha sido la siguiente:

1. Para la preparación de una asignatura, ¿Prefiere realizar ejercicios con un juego o mediante un sistema informático de ejercicios tradicional sin juego?
   a) Prefiero el sistema de juego
   b) Prefiero el sistema tradicional

2. Para la evaluación de una asignatura, ¿Prefiere realizar ejercicios con un juego o mediante un sistema informático de ejercicios tradicional sin juego?
   a) Prefiero el sistema de juego
   b) Prefiero el sistema tradicional

Por favor, puntúe de 1 (nada de acuerdo) a 7 (totalmente de acuerdo), las siguientes afirmaciones

Afirmaciones	Evaluación							
	1	2	3	4	5	6	7	NC
El uso de un juego para resolución de ejercicios me parece adecuado para la evaluación de una								

asignatura								
El hecho de realizar un juego me motiva más								
El hecho de realizar una competición en el juego me puede ocasionar sentimientos negativos								
El uso de un juego para resolución de ejercicios me parece adecuado para la preparación de una asignatura								

Tabla 4.1. Tabla preguntas pre-encuesta

Con la segunda encuesta se quiere saber si el alumno ha cambiado de opinión respecto a los juegos educativos y conocer su opinión general del juego. Estas son las preguntas formuladas después de jugar:

1. Para la preparación de una asignatura, ¿Prefiere realizar ejercicios con un juego o mediante un sistema informático de ejercicios tradicional sin juego?
    a) Prefiero el sistema de juego
    b) Prefiero el sistema tradicional

2. Para la evaluación de una asignatura, ¿Prefiere realizar ejercicios con un juego o mediante un sistema informático de ejercicios tradicional sin juego?
    a) Prefiero el sistema de juego
    b) Prefiero el sistema tradicional

Por favor, puntúe de 1 (totalmente en desacuerdo) a 7 (totalmente de acuerdo), las siguientes afirmaciones

Preguntas	Evaluación							
	1	2	3	4	5	6	7	NC
Me ha gustado y estoy satisfecho con la experiencia realizada con el juego								

Me gustaría repetir en asignaturas reales la experiencia con el juego								
Me ha gustado el juego utilizado								
Creo haber aumentado mi aprendizaje de la asignatura durante la experiencia								
El uso del juego para resolución de ejercicios me parece útil para la preparación de la asignatura								
El uso del juego para resolución de ejercicios me parece adecuado para la evaluación de la asignatura								
El puesto en el que he quedado en la clasificación me parece justo								
La utilización de este juego me ha motivado								
El hecho de realizar una competición en el juego me puede ocasionar sentimientos negativos								
La interacción con el sistema informático del juego ha sido sencilla (facilidad de entender los menús, contestar las preguntas, etc.)								
El sistema de ganar dinero me ha parecido justo								

Tabla 4.2. Tabla afirmaciones post-encuesta

Por favor, indique los tres aspectos de la experiencia que considera más positivos

Por favor, indique los tres aspectos de la experiencia que considera más negativos

4.2.-Resultados

Ahora se detallan los resultados obtenidos de las evaluaciones de la sección 4.1, tanto de las tablas 4.1. y 4.2 como de las preguntas.

Se ha calculado la media y varianza de las muestras obtenidas. Hay que tener en cuenta que los resultados de las tablas 4.4 y 4.6 se han calculado en base a un valor máximo de siete.

Resultados de las encuestas realizadas antes de jugar:

	1.- Para la preparación de una asignatura, ¿Prefiere realizar ejercicios con un juego o mediante un sistema informático de ejercicios tradicional sin juego?	2.- Para la evaluación de una asignatura, ¿Prefiere realizar ejercicios con un juego o mediante un sistema informático de ejercicios tradicional sin juego?
Prefiero el sistema de juego	**8**	**6**
Prefiero el sistema tradicional	**2**	**4**

Tabla 4.3. Tabla preguntas pre-encuesta

Afirmaciones	Evaluación									
	1	2	3	4	5	6	7	NC	media	varianza
El uso de un juego para resolución de ejercicios me parece adecuado para la evaluación de una asignatura	0	0	1	0	0	7	2	0	5.9	1.09
El hecho de realizar un juego me motiva más	1	1	2	1	2	2	1	0	4.2	3.36
El hecho de realizar una competición en el juego me puede ocasionar sentimientos negativos	0	0	1	0	0	3	6	0	6.3	1.41
El uso de un juego para resolución de ejercicios me parece adecuado para la preparación de una asignatura	1	2	2	2	1	1	1	0	3.7	3.21

Tabla 4.4. Tabla evaluación afirmaciones pre-encuesta

Resultados de la encuesta después de jugar a shopC:

	1.- Para la preparación de una asignatura, ¿Prefiere realizar ejercicios con un juego o mediante un sistema informático de ejercicios tradicional sin juego?	2.- Para la evaluación de una asignatura, ¿Prefiere realizar ejercicios con un juego o mediante un sistema informático de ejercicios tradicional sin juego?
Prefiero el sistema de juego	**8**	**5**
Prefiero el sistema tradicional	**2**	**5**

Tabla 4.5. Tabla evaluación preguntas post-encuesta

Preguntas			Evaluación							
	1	2	3	4	5	6	7	NC	**media**	**varianza**
Me ha gustado y estoy satisfecho con la experiencia realizada con el juego	**0**	**0**	**0**	**1**	**1**	**6**	**2**	**0**	**5.9**	**0.69**
Me gustaría repetir en asignaturas reales la experiencia con el juego	**0**	**0**	**1**	**1**	**1**	**5**	**2**	**0**	**5.6**	**1.44**
Me ha gustado el juego utilizado	**0**	**0**	**0**	**1**	**4**	**5**	**0**	**0**	**5.4**	**0.44**
Creo haber aumentado mi aprendizaje de la asignatura durante la experiencia	**0**	**0**	**1**	**2**	**3**	**3**	**1**	**0**	**5.1**	**1.29**
El uso del juego para resolución de ejercicios me parece útil para la	**0**	**0**	**1**	**1**	**1**	**6**	**1**	**0**	**5.5**	**1.25**

preparación de la asignatura										
El uso del juego para resolución de ejercicios me parece adecuado para la evaluación de la asignatura	**1**	**0**	**0**	**2**	**2**	**5**	**0**	**0**	**4.9**	**2.29**
El puesto en el que he quedado en la clasificación me parece justo	**0**	**0**	**1**	**4**	**4**	**0**	**1**	**0**	**4.6**	**1.04**
La utilización de este juego me ha motivado	**0**	**0**	**2**	**1**	**3**	**3**	**1**	**0**	**5.0**	**1.6**
El hecho de realizar una competición en el juego me puede ocasionar sentimientos negativos	**3**	**3**	**0**	**0**	**2**	**2**	**0**	**0**	**3.1**	**4.06**
La interacción con el sistema informático del juego ha sido sencilla (facilidad de entender los menús, contestar las preguntas, etc.)	**0**	**0**	**0**	**1**	**5**	**3**	**1**	**0**	**5.4**	**0.64**
El sistema de ganar dinero me ha parecido justo	**0**	**0**	**2**	**2**	**3**	**1**	**2**	**0**	**4.9**	**1.89**

Tabla 4.6. Tabla evaluación afirmaciones post-encuesta

Tras ver las encuestas realizadas se obtienen las siguientes conclusiones:

- A la mayoría de alumnos, un 80%, prefiere los juegos educativos para la preparación de las asignaturas y tras el juego se mantienen los porcentajes.
- Un porcentaje menor, un 60% prefiere los juegos para la evaluación ante un 40% que no, aunque después de jugar a *shopC,* el porcentaje se iguala a un 50%. Observar tablas 4.3 y 4.5.

- En gran proporción los alumnos creen que un juego les podría motivar más en las clases y las asignaturas.
- La mitad de los encuestados creen que un juego les puede ocasionar sentimientos negativos.

Respecto a este juego en concreto:

- En general no se ha cambiado la buena opinión inicial respecto a los juegos educativos.
- Los estudiantes quedaron satisfechos con las reglas del juego, excepto una chica que perdió y cree que es injusta la puntuación y muy difíciles las preguntas.
- Casi todos los que ha jugado han terminado o están terminando la carrera por lo que la asignatura de Arquitectura de Sistemas ya la aprobaron hace algunos años y piensan que ha sido una buena forma de recordar lo estudiado.
- A todos les ha gustado la interfaz.
- Piensan que las reglas del juego son claras y sencillas, en general.
- Sobre la dificultad de las preguntas hay diversas opiniones, algunos creen que han sido difíciles y otros que normales, nadie cree que hayan sido fáciles.
- Los aspectos negativos son:
  - Muchos piensan que el juego debería ser online, para poder jugar cada uno desde su casa y no todos desde el mismo ordenador.
  - Les gustaría que tuviera más dados, más jugadores, más azar mediante tarjetas o algo parecido, más casillas, etc.
  - Creen que tener sonido lo haría más entretenido.
  - Les gustaría que hubiese más animaciones.
  - Piensan que sería mejor que se pudiesen guardar más de cuatro partidas.
  - Que se pudieran revender o hipotecar las tiendas.
  - Si juegas varias partidas se repiten las preguntas.

- Los aspectos positivos son:
  - La corrección de las preguntas.
  - Han recordado conceptos sobre la asignatura de Arquitectura de Sistemas.
  - Creen que motivaría para estudiar.
  - Entretenido y divertido.
  - Que se pueden guardar las jugadas.
  - Las interfaces bonitas.
  - Reglas del juego sencillas y lógicas.

5. Futuras líneas:

Una posible ampliación es añadir al juego más funcionalidades:

- Tarjetas, para que cuando un jugador caiga en determinadas casillas pueda coger una tarjeta y tenga que hacer lo que en ella ponga.
- Hacer unos recreativos o sala de juegos en el centro comercial y al caer aquí el jugador levantará una tarjeta, en las tarjetas puede poner: “El jugador ha perdido 5M. jugando al billar” ó “El jugador ha ganado 15M. en el bingo”.
- Implementar que en lugar de perder quién se quede sin dinero pudiera revender sus propiedades y así conseguir dinero para continuar jugando ó que contestando bien a ciertas preguntas, al caer en una casilla ya comprada, ese jugador pudiera quitar la tienda al que primero la compró por cierto dinero.
- Sería interesante que en el juego de forma aleatoria saliera un “vigilante de seguridad” y diera órdenes o castigos al jugador que en ese momento tenga el turno.
- Más jugadores.
- Más casillas.
- Un tiempo para contestar a las preguntas.
- Hacer el juego online.
- Más dados.
- Sonidos.
- Más animaciones.
- Poder guardar más partidas.

Incluso otra ampliación, más costosa, podría ser implementar el juego para dispositivos Android o IOS. También se podría extender la evaluación para medir las ganancias de aprendizaje y hacerlo un número mayor de alumnos.

6. Presupuesto y planificación

6.1. Presupuesto:

En este apartado se justifican los costes derivados de la realización de este proyecto desde el momento en que se inició hasta el final, la redacción de la memoria.

La duración del trabajo ha sido de un año, desde Febrero de 2012 hasta el mismo mes del año siguiente.

La primera época del trabajo, desde Febrero hasta Junio de 2012 se compatibilizó con las últimas asignaturas de la carrera y el resto de meses con un trabajo de 5 horas diarias.

La tabla 6.1 resume el trabajo realizado y las horas estimadas que ha llevado la realización de esta aplicación:

TRABAJO	TIEMPO	MESES
Repaso del lenguaje Java	5 horas	Febrero 2012 y Marzo 2012
Ideas y especificación de requisitos.	30 horas	Marzo 2012 y Mayo 2012
Instalación de Eclipse	1 hora	Mayo 2012
Instalación de librerías Java	2 horas	Junio 2012
Desarrollo del funcionamiento aplicación	170 horas	Julio 2012 y Agosto 2012
Estudio de MySQL	10 horas	Septiembre 2012
Instalación XAMPP para el uso MySQL	2 horas	Septiembre 2012
Conexión MySQL y Java	10 horas	Octubre 2012
Creación de la base de datos y sus tablas	10 horas	Octubre 2012 y Noviembre 2012
Realización de interfaz	50 horas	Diciembre 2012 y Enero 2013
Redacción de la Memoria	135horas	Diciembre 2012, Enero 2013 y Febrero 2013
TOTAL:	425 horas	

Tabla 6.1. Planificación

Si se establece un precio por hora de 17€, el coste por el trabajo personal sería de 7.225€ en total, ya que el total de horas que me ha llevado la realización del proyecto es de 425 horas. En la tabla 6.2 se desglosan los costes materiales.

HERRAMIENTAS	PRECIO	AMORTIZACIÓN	IMPORTE
Portátil Sony vaio	800€	1/4	200€
Documentación	-	-	100€
Photoshop CS6	942,82€	1/4	235,71€
	TOTAL:	535,71€	

Tabla 6.2. Costes y materiales.

Algunos materiales sufren desgaste por amortización, por lo que usaré un coeficiente de amortización de ¼, el portátil y el programa Photoshop. El total de costes incrementa en 535,71€.

Por lo tanto, incluyendo los costes indirectos, que serán 700€, y un 21% de I.V.A, el precio total será de 9.442,25€, **nueve mil cuatrocientos cuarenta y dos euros con veinticinco céntimos.**

La distribución de horas y de precios es aproximada ya que es muy difícil decir con exactitud el tiempo que he pasado con el TFG y el precio exacto de las herramientas que he utilizado.

6.2.-Planificación:

En la tabla 6.3 se desglosan las tareas realizadas y las horas que me han llevado aproximadamente:

TAREA	DESCRIPCIÓN	HORAS
1	**Repaso del lenguaje Java**	**5 horas**
1.1	Buscar y agrupar documentación sobre Java	1 horas
1.2	Leer documentación sobre Java	4 horas
2	**Ideas e informe inicial sobre la aplicación**	**30 horas**
2.1	Ideas para la aplicación	25 horas
2.2	Redacción de especificaciones	5 horas
3	**Instalación de Eclipse**	**1 hora**
3.1	Búsqueda e instalación de Eclipse	1 hora
4	**Instalación de librerías Java**	**2 horas**
4.1	Búsqueda y descarga de los JDK	1 hora

4.2	Instalación y pruebas de funcionamiento de los JDK	1 hora
5	**Desarrollo del funcionamiento aplicación**	**170 horas**
5.1	Idear la estructura de la aplicación	5 horas
5.2	Programar la funcionalidad de la aplicación	80 horas
5.3	Programar la interfaz de usuario	80 horas
5.4	Unión de funcionalidad e interfaz	5 horas
6	**Estudio de MySQL**	**10 horas**
6.1	Búsqueda de documentación sobre MySQL	3 horas
6.2	Leer documentación sobre MySQL	7 horas
7	**Instalación XAMPP para el uso MySQL**	**2 horas**
7.1	Entender el uso de XAMPP	1 horas
7.2	Descargar e instalar XAMPP	1 hora
8	**Conexión MySQL y Java**	**10 horas**
8.1	Buscar documentación para conectar MySQL a la aplicación	2 horas
8.2	Leer y comprender documentación de conexión	3 horas
8.3	Programar Java para conectar mi aplicación con MySQL	5 horas
9	**Creación de la base de datos y sus tablas**	**10 horas**
9.1	Aprender y crear una base de datos MySQL desde una aplicación Java	3 horas
9.2	Aprender y crear tablas en una base de datos MySQL desde una aplicación Java.	7 horas
10	**Realización de interfaz**	**50 horas**
10.1	Diseñar imágenes para la interfaz de la aplicación	10 horas
10.2	Información sobre programas de edición: Photoshop, GIMP 2 y Paint.	5 horas
10.3	Creación de imágenes para la interfaz de la aplicación	25 horas
10.4	Incorporación de las imágenes a la aplicación	10 horas
11	**Redacción de la Memoria**	**135 horas**
11.1	Estructura interna de la memoria (índice)	5 horas
11.2	Búsqueda y agrupación de recursos	35 horas
11.3	Redacción de la memoria	90 horas
11.4	Estructura externa de la memoria (portada, imágenes…)	5 horas

Tabla 6.3. Tareas.

7. Conclusiones

En este trabajo se ha diseñado e implementado un juego de tablero por ordenador denominado *shopC* para la ayuda en el aprendizaje. El juego se ha implementado utilizando el lenguaje de programación Java y está basado en un centro comercial en el cual se pueden comprar tiendas y recibir dinero cuando los jugadores caen en las propiedades del propietario.

En el juego se han integrado elementos instruccionales para el aprendizaje, como son las preguntas educativas a formular a los participantes, con elementos de interfaz gráfica, y con las reglas del juego de tablero basado en el centro comercial. Entre las diferentes opciones posibles de integración, se ha intentado seleccionar una que al mismo tiempo resultara justa en cuanto a la evaluación educativa, también resultara divertida y motivante para los participantes.

Aparte de los diferentes tests para verificar la funcionalidad del software, este juego educativo ha sido evaluado con 10 alumnos con preguntas cargadas sobre la asignatura de Arquitectura de Sistemas. Los alumnos han contestado a varias encuestas antes y después de utilizar la herramienta. En base a los resultados de las encuestas se puede establecer que se han conseguido los objetivos propuestos inicialmente:

- Las interfaces creadas son bonitas y sencillas.
- Las reglas del juego son lógicas y entretenidas para los alumnos.
- Los estudiantes han aprendido y han recordado conceptos de la asignatura Arquitectura de Sistemas, por lo tanto se puede decir que han aumentado y reforzado sus conocimientos.
- Los evaluados se han sentido muy motivados con el juego y les ha parecido divertido y entretenido.
- Para los encuestados es positivo que se puedan guardar jugadas para reanudarlas después.

- Fruto de las pruebas y evaluaciones realizadas, el programa software ha ido evolucionando, para adaptarse mejor a los objetivos. Así por ejemplo, la regla que fija la terminación del juego ha cambiado respecto al inicio, teniendo finalmente un límite de 50 turnos o que un jugador se quede sin dinero, ya que la regla inicial de que el juego no terminase hasta que todos menos uno se quedara sin dinero, hacia el juego en ocasiones muy largo y podía ser aburrido.
- Los objetivos personales también han sido conseguidos:

- Se ha mejorado la programación en Java, conociendo mejor su sintaxis, sus clases y sus librerías; lo que ha permitido desarrollar y establecer la lógica de la aplicación.
- Se ha aprendido en profundidad a utilizar el gestor de bases de datos MySQL que ha permitido dar mucha funcionalidad al juego.
- Se ha logrado manejar con soltura Eclipse que facilita mucho la programación.
- Se ha mejorado el uso de programas de diseño, tales como Paint, Photoshop, GIMP 2, etc.

Bibliografía

[1] P. J. Muñoz, F. Martínez, M. Muñoz y C. Delgado Kloos, *MagicLearning: A Serious Game for Learning based in a Magic World,* ICIC International ©2011 ISSN 1349-4198, pp. 1-13, 2011.

[2] E. Thorndike*, Education: A first book*, Macmillan, 1912.

[3] R. Provine, Universidad de Maryland, http://www.rtve.es/tve/b/redes2007/semanal/prg361/entrevista.htm

[4] M. A. Andreu y M. García, *Actividades lúdicas en la enseñanza de LFE:el juego didáctico.*

[5] J. Huizinga, *Homo Ludens*, Madrid: Alianza, 1984.

[6] Real Academia Española, *Diccionario de la lengua española* (22.aed.), 2001 Consultado en http://www.rae.es/rae.html

[7] T. Butler, Games and Simulations: Creative Educational Alternati- ves, Bch Dends, Volumen 33, 1988.

[8] V. Viciana, J.L. Conde y J.Conde, *El juego como vehículo para la adquisión de los aprendizajes*, 2002.

[9] R. Ortega & T. Lozano, *Espacios de juego y desarrollo de la autonomía y la identidad en la Educación Infantil*. Revista Aula Nº 52-52, pp. 13-17, 1996.

[10] M. De Borja, *Situación Española, Experiencias de Juego en Preescolares*, 1985.

[11] C. Giménez, C. Pagés y J.J. Martínez, *Diseño y desarrollo de un juego educativo para ordenador sobre enfermedades tropicales y salud internacional: una herramienta docente más de apoyo al profesor universitario,* Revista Eureka sobre Enseñanza y Divulgación de las Ciencias 8 (2), pp. 221-228, 2011.

[12] Z. Chen, C. Y. Liao, and T. Chan, Learning by Pet-training Competition: Alleviating Negative Influences of Direction Competition by Training Pets to Compete in Game-based Environments, *Proc. of the 10th IEEE International Conference on Advanced Learning Technologies*, Sousse, pp. 411-413, 2010.

[13] C. Lin, S.S. Young, and H. Hung, The Game-based Constructive Learning Environment to Increase English Vocabulary Acquisition: Implementing a Wireless Crossword Fan-Tan Game (WiCFG) as an example, *Proc. of the 5th IEEE International Conference on Wireless, Mobile, and Ubiquitous Technology in Education*, Beijing, pp.205-207, 2008.

[14] L. Chang, J. Yang, and F. Yu, Development and evaluation of multiple competitive activities in a synchronous quiz game system, *Innovations in Education & Training International*, vol.40, no.1, pp.16-26, 2003.

[15] J. Byous*, Java technology: The early years. Sun Developer Network*, Recuperado en 2005.

[16] J. Gosling, A brief history of the Green project. Java.net, Recuperado en 2005.

[17] Eclipse Membership www.eclipse.org

[18] Wikipedia, la enciclopedia libre, http://es.wikipedia.org/wiki/MySQL

[19] A. Beaulieu, *Learning SQL*, Second Edition, O'Reilly Media, 2009.

[20] M. Domínguez, *Todo Programación*. Nº 6-7. Págs. 35-38. Editorial Iberprensa (Madrid). *Acceso a bases de datos desde aplicaciones Java: JDBC 1.0.*, 2004.

[21] Sun Microsystems Inc., *JavaHelp TM 2.0 System User's Guide,* 2004.

ANEXO 1: Manual de usuario

La base fundamental de este juego consiste en la adquisición de tiendas de este centro comercial y así conseguir dinero desbancando a los demás jugadores. Todo esto con la dificultad añadida de que para poder adquirir propiedades hay que demostrar cultura e inteligencia contestando correctamente ciertas preguntas.

>>REGLAS

Cada uno de los jugadores tendrá un panel como el de la figura a1.2 que contiene su nombre, que lo pone al principio de la jugada, su color de ficha, el dinero que tiene para comprar las propiedades y una lista de la compra donde se van guardando las propiedades que el jugador va comprando.

Figura a1.1 Elementos panel de cada jugador

A la salida cada jugador recibirá una cantidad inicial de dinero (detallada más abajo) y comenzará a mover el Jugador1, después el Jugador2 y así sucesivamente. Se empezará desde la casilla de Parking.

El jugador que tenga el turno deberá tirar el dado pulsando encima de él. El número que salga será el número de casillas que avanzará. Las fichas se moverán siempre en el sentido de las agujas del reloj.

Durante toda la partida las fichas quedarán en las casillas o tiendas correspondientes hasta que avancen a otra por una nueva tirada del dado, pudiendo contener una misma casilla o tienda más de una ficha.

Figura a1.2 Elementos del tablero de juego

En la figura a1.2 se indican los algunos de los elementos del tablero. Como son la posición inicial de la fichas, la máquina de turnos, que indica el turno de cada jugador y el dado.

Si la ficha cae en una propiedad que no esté comprada, aparecerá una venta preguntado si quiere comprar esa tienda.

Si no se quiere comprar, bastará con pulsar el botón *cancelar*.

Si se quiere comprar la tienda, se pulsará *comprar* y automáticamente aparecerá una ventana con tres tipos de preguntas a las que el jugador deberá contestar. Pagará dependiendo de las preguntas acertadas:

- Acierta todas las preguntas →paga un 25% del valor de la propiedad.
- Acierta dos preguntas →paga un 50% del valor de la propiedad.
- No acierta ninguna pregunta →paga el 100% del valor de la propiedad.

Tanto si acierta como si no, el turno pasará al siguiente jugador.

Si un jugador cae en una tienda comprada por él con anterioridad recibirá un 25% del precio inicial de la tienda.

Si por el contrario, un jugador cae en una propiedad que ya está comprada por otro jugador, se le formulará una pregunta:

- Si acierta, no pagará nada.
- Si no acierta, perderá un 50% del valor de la propiedad.

>>DINERO INICIAL DE CADA JUGADOR:

En la repartición inicial de dinero se tendrá en cuenta que el centro comercial completo está valorado en 600M. Por lo tanto dependiendo de los jugadores que participen se le dará una cantidad de dinero inicial u otra. Cuántos más jugadores, menos dinero recibirán.

- Cuatro jugadores →150M. a cada jugador.
- Tres jugadores →200M. a cada jugador.
- Dos jugadores →300M. a cada jugador.
- Un jugador →600M.

>>TIPOS DE CASILLAS:

Para obtener información de cada tienda o casilla bastará con pulsar con el ratón encima de está y aparecerá una ventana informativa.

Hay distintos tipos de casillas en el tablero de este juego, se explican a continuación:

CASILLAS PARA COMPRAR:

Son todas las casillas donde hay una tienda o un restaurante, ver figura a1.2. El jugador que caiga en una de estas casillas podrá comprarla si no está comprada respondiendo a tres preguntas, el precio dependerá de los aciertos. Está explicado con más detalle anteriormente.

CASILLAS DE BAÑO:

Son todas las casillas de la esquina superior derecha y de la esquina inferior izquierda del tablero.

Cuando un jugador caiga aquí, tendrá dos opciones y tendrá que elegir una para continuar:

- Pagar 5M. sólo por usar los baños y perderá el turno que pasará al siguiente jugador.
- Pagar 10M. por entrar a los baños y fastidiar al siguiente jugador quitándole su turno para jugar.

CASILLA DE PARKING:

Es la primera casilla del tablero, donde se inicia el juego. El comprador dejará su coche aquí aparcado y cada vez que pare en está casilla deberá pagar 10M. por alargar su estancia en el Centro Comercial. Pagará y el turno pasará al siguiente jugador.

CASILLA DE INFORMACIÓN:

Si un jugador cae en esta casilla será premiado por el centro comercial con un cheque de 10M. y pierde su turno.

ESCALERAS MECÁNICAS:

Hay dos tipos de escaleras, unas solo de subida y otras solo de bajada.

Dependiendo de por cuál de las dos pase, a el jugador se le formulará la pregunta y podrá decidir si quiere subirlas ó bajarlas, es decir, ir de las tiendas de la parte superior del tablero a las de la parte inferior, si pasa por las escaleras de bajada y a la inversa si pasa por las escaleras de subida. Tanto si decide usar las escaleras cómo sino, podrá comprar la Tienda donde caiga.

>>OBJETIVO

El objetivo de juego es no quedarse sin dinero, en el momento que un jugador tenga menos de 0M. el juego finaliza siendo este el perdedor de la partida.

El ganador será aquél jugador que tenga más dinero acumulador, sin tener en cuenta el valor de las propiedades que posea en el momento de acabar el juego.

El juego también acaba cuando pasen 50 turnos. Se hará un recuento solamente del dinero acumulado, sin tener en cuentas las Tiendas compradas, y el jugador que menos dinero en total tenga será el perdedor. El ganador será el que más dinero posea.

ANEXO 2: Manual del profesor

El profesor debe ser capaz de crear nuevas preguntas y editar las existentes. Para acceder a la base de datos y modificar las tablas de las preguntas, se debe hacer lo siguiente:

Entrar en MySQL y abrir la base de datos con nombre test

a. Nombre usuario: root
b. Sin contraseña.

La línea en código SQL sería la siguiente:

```
mysql>use test
```

1. Las tablas que hay en la base de datos test se podrán mostrar si a continuación pone : `mysql>show tables`
   Y le aparecerá una tabla como en la figura a2.1:

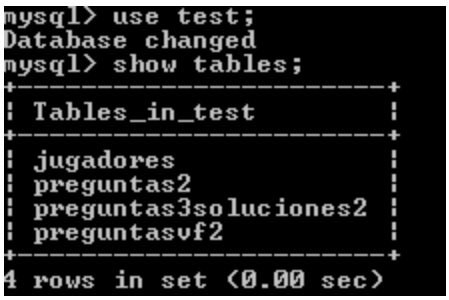


Figura a2.1 Cómo mostrar las tablas de una base de datos

La tabla jugadores, es la que guarda las partidas.

La tabla preguntas2 es la tabla con las preguntas y respuestas de responder en hueco.

La tabla preguntas3soluciones contiene las preguntas, las tres opciones a respuesta y la solución.

La tabla preguntasvf2 almacena las preguntas y las respuestas de Si o No.

2. Para ver lo que contienen las tablas puede hacer dos cosas:
   a. Si pone: `mysql>describe nombretabla`
   Podrá ver los tipos y campos de la tabla.
   En la figura a2.2 se observa una de las tablas con sus campos.

```
mysql> describe preguntas2;
+-----------+---------------+------+-----+---------+-------+
| Field     | Type          | Null | Key | Default | Extra |
+-----------+---------------+------+-----+---------+-------+
| id        | int(10)       | YES  |     | NULL    |       |
| pregunta  | varchar(5000) | YES  |     | NULL    |       |
| respuesta | varchar(5000) | YES  |     | NULL    |       |
+-----------+---------------+------+-----+---------+-------+
3 rows in set (0.20 sec)
```

Figura a2.2 Cómo ver los campos de una tabla con SQL.

c. Si pone: `mysql>select * from nombretabla`

Podrá ver el contenido de las tablas. Si utiliza '*' verá la tabla entera y si en su lugar pones un campo de la tabla solo se mostrará el contenido de ese campo.

En la figra a2.3 se muestra el contenido de la tabla al completo.

```
mysql> select * from preguntas2;
+------+------------------------------------------------------------------------------
| id   | pregunta
+------+------------------------------------------------------------------------------
|    1 | Un parßmetro que se pasa a una funci¾n es una variable
|    2 | Si una variable tiene que almacenar la direcci¾n de la direcci¾n de un carßcter, ┐quú tipo d
|    3 | ┐Se puede imprimir la direcci¾n de memoria donde estß guardada una cadena?,
|    4 | La funci¾n que, dado el nombre de un fichero, lo borra del sistema de ficheros:
|    5 | La funci¾n, que renombra un ficher
|    6 | Fichero nuevo, new.c y quieres subirlo al repositorio remoto. ┐Cußl serÝa la secuencia de co
|    7 | ┐ svn --username 100011111 commit new.c es lo mismo que svn --username 100011111 add new.c ?
|    8 | ┐El c¾digo de un programa escrito en C se divide en funciones o mÚtodos?
```

Figura a2.3 Cómo ver los contenidos de una tabla con SQL.

En la figura a2.3 se muestra como tan solo se puede ver el campo "id":

```
mysql> select id from preguntas2;
+-------+
| id    |
+-------+
|     1 |
|     2 |
|     3 |
|     4 |
|     5 |
|     6 |
|     7 |
|     8 |
|     9 |
|    10 |
```

Figura a2.3 Mostrar una columna en concreto de cierta tabla en SQL.

>>INSERTAR PREGUNTA.

Para insertar una pregunta en una tabla se debe utilizar *INSERT INTO*, el nombre de la tabla y en *VALUES* los valores se quiere insertar en cada una de las columnas. Por ejemplo si quiero insertar una pregunta y su solución en la tabla llamada preguntas2, haría lo siguiente:

```
mysql> INSERT INTO preguntas2 VALUES ('NumeroId', 'pregunta', 'respuesta')
```

Es muy importante que el número *id* no esté repetido.

>>BORRAR PREGUNTA.

Para borrar una pregunta de la tabla llamada *preguntas2*, quiero borrar el elemento de la columna *id* con número *NumeroId*:

```
mysql> DELETE FROM preguntas2 WHERE id='NumeroId'
```

ANEXO 3: Manual de instalación

Para que funcione esta aplicación en cualquier ordenador deben seguirse los siguientes pasos:

Primero se deberá instalar MySQL en el ordenador. Pedirá un usuario y contraseña y para que funcione la aplicación, el usuario deberá ser root y no se debe poner contraseña. No hace falta crear ninguna base de datos ya que se utiliza una que se crea por defecto al instalar MySQL. Asegúrese que MySQL está arrancado y que su puerto, que suele ser 3306, no está siendo utilizado.

Para arrancar y parar MySQL se aconseja utilizar XAMPP, que se instala fácilmente en el ordenador y es muy sencillo de usar.

Segunda y última cosa que ha de instalar es una máquina virtual Javapara que el ejecutable funcione correctamente.

Una vez esté todo instalado se podrá jugar haciendo doble clic en el archivo jar o ejecutable.